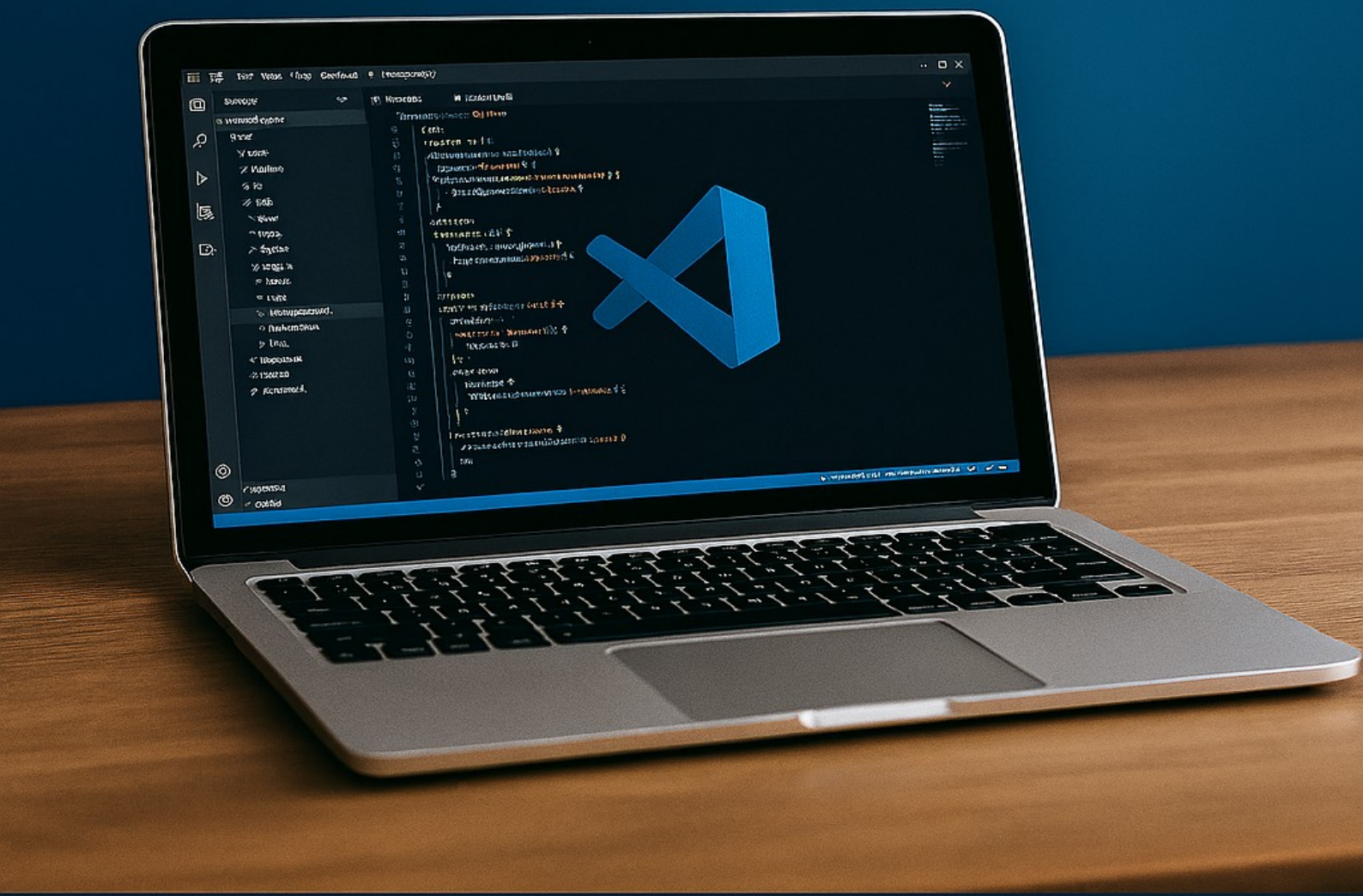
VISUAL STUDIO CODE

MANUAL



VISUAL STUDIO CODE

manual

INDICE

I. Introducción y Preparación para VS Code A. Por qué aprender VS Code B. Requisitos Conclusión de la lección

II. Descarga e Instalación de VS Code A. Descarga B. Instalación en Windows C. Primera Vista Conclusión de la lección

III. Configuración Inicial (Idioma) A. Idioma por defecto (Inglés) B. Cambio de idioma a Español C. Revertir a Inglés Conclusión de la lección

IV. Componentes Principales de la Aplicación A. Barra de Actividades (Activity Bar) B. Barra de Estado (Status Bar) C. Menú Superior Conclusión de la lección

V. Uso del Explorador de Archivos (Explorer) A. Acceso B. Apertura de Carpeta C. Botones del Explorer D. Creación y Edición de Archivos E. Organización de Archivos F. Manipulación de Texto Conclusión

VI. Funcionalidades de la Barra de Estado (Status Bar) A. Indicadores Izquierdos B. Posición del Cursor C. Configuración de Identación (Indentation) D. Codificación y Final de Línea E. Tipo de Archivo / Lenguaje Conclusión

VII. Búsquedas y Reemplazos A. Acceso B. Búsqueda General C. Opciones de Búsqueda D. Reemplazo de Texto Conclusión

VIII. Control de Versiones (Git) A. Instalación de Git B. Inicialización del Repositorio C. Configuración de Usuario D. Conexión con GitHub y Publicación E. Manejo de Ramas (Branches) Conclusión

IX. Depuración (Debugger) A. Requisitos y Obtención del Proyecto B. Inicio del Debugger C. Uso de Breakpoints D. Componentes de la Interfaz del Debugger E. Controles de Depuración F. Consola de Depuración (Debug Console) Conclusión

X. Extensiones y Temas A. Funcionalidad de Extensiones B. Temas C. Iconos D. Cambio de Tema Rápido Conclusión

XI. Atajos de Teclado y Edición Rápida A. Navegación B. Manipulación de Líneas C. Indentación D. Comentarios Conclusión

XII. IntelliSense y Snippets A. IntelliSense B. Creación de Snippets (Fragmentos de código) C. Snippets de Terceros Conclusión

XIII. Navegación Rápida y Paleta de Comandos A. Búsqueda de Archivos ($Control+P$) B. Paleta de Comandos ($Shift+Control+P$) Conclusión

XIV. Personalización del Layout de VS Code A. Mover la Barra Lateral Primaria (Sidebar) B. Ocultar la Barra de Actividades (Activity Bar) C. Alternar visibilidad de la barra lateral D. Ocultar pestañas (Tabs) E. Navegación entre archivos abiertos sin pestañas Conclusión

XV. Atajos de Teclado Customizados A. Enfocar el Explorador B. Creación de un Atajo Customizado: Crear Nuevo Archivo con la tecla 'a' Conclusión

XVI. Configuración de la Terminal Integrada A. Cambio de Terminal por Defecto B. Abrir la Terminal Integrada Conclusión

XVII. Conclusión A. Finalización del curso y logros B. Agradecimiento y llamados a la acción Cierre motivacional

Guía Rápida de VS Code

I. Introducción y Preparación para VS Code

A. Por qué aprender VS Code

Visual Studio Code, o simplemente VS Code, es un editor de código fuente creado por Microsoft que se ha convertido en una de las herramientas más amadas por desarrolladores de todo el mundo. Antes de empezar a usarlo, es importante comprender por qué merece la pena dominarlo.

1. **Es el editor de código más popular en la actualidad.**
   Desde su lanzamiento en 2015, VS Code ha ganado una enorme comunidad de usuarios y ha sido votado repetidamente como el editor más utilizado en las encuestas anuales de *Stack Overflow.* Su popularidad no es casual: combina potencia, ligereza y versatilidad. Puedes escribir en él desde simples scripts en Python hasta proyectos complejos de desarrollo web, aplicaciones móviles o software multiplataforma.

   Además, la comunidad global mantiene un flujo constante de extensiones, temas, complementos y mejoras que hacen que siempre esté evolucionando. Aprender a usarlo te coloca en sintonía con el estándar actual del desarrollo moderno.

2. **Corre en cualquier sistema operativo.**
   VS Code está disponible para Windows, macOS y Linux. No importa en qué sistema trabajes, siempre tendrás la misma experiencia, interfaz y comportamiento.
   Esto permite que los equipos de desarrollo trabajen de manera uniforme sin importar si uno usa una Mac y otro un PC con Ubuntu. Solo necesitas descargar el instalador apropiado desde la página oficial y podrás comenzar en minutos.

3. **Es de código abierto y completamente gratis.**
   Una de las mayores ventajas de VS Code es que no necesitas pagar licencias ni suscripciones. Está publicado como software libre bajo la licencia MIT, lo que significa que cualquier persona puede usarlo, modificarlo o incluso contribuir al desarrollo del proyecto. Esto también asegura transparencia: puedes ver el código fuente completo en GitHub y entender cómo funciona internamente, lo que da confianza y abre oportunidades de aprendizaje profundo para quienes quieren explorar cómo se construye un editor profesional.

4. **Permite editar y escribir código muy rápido con sugerencias de autocompletado.**
   Una de las joyas de VS Code es su *IntelliSense*, un sistema de autocompletado inteligente que analiza tu código y te sugiere variables, funciones, etiquetas o métodos en tiempo real. Esta característica reduce errores, acelera el flujo de trabajo y hace que programar sea mucho más fluido. Además, puedes instalar extensiones específicas para cada lenguaje (por ejemplo, Python, C#, JavaScript, C++, etc.), que amplían las sugerencias y el resaltado de sintaxis para adaptarse perfectamente a lo que estés programando.

   Por si fuera poco, puedes navegar fácilmente entre archivos, buscar funciones, depurar código y ejecutar scripts sin salir del entorno, lo que convierte a VS Code en una auténtica “central de control” para cualquier desarrollador.

5. **Es personalizable (se puede dejar “sumamente sensual”).**
   Una de las razones más divertidas para aprender VS Code es su grado de personalización. Puedes cambiar temas de color, íconos, tipografías y disposición de ventanas, creando un entorno de trabajo que refleje tu personalidad y te inspire a programar.

Existen miles de temas creados por la comunidad, desde los más minimalistas hasta los más coloridos. Además, puedes añadir extensiones para animaciones, barras de estado, sonidos o fondos dinámicos.
Tener un editor visualmente atractivo y ordenado mejora la concentración, hace el trabajo más agradable y potencia la creatividad. En resumen, puedes convertir tu VS Code en una herramienta “sumamente sensual” a tu gusto.

B. Requisitos

Antes de comenzar a instalar y explorar Visual Studio Code, asegúrate de cumplir con algunos requisitos básicos para sacar el máximo provecho de la experiencia.

1. **Un computador.**
   Puede ser un ordenador portátil o de escritorio, con cualquier sistema operativo (Windows, macOS o Linux). VS Code es liviano y no necesita un hardware de alto rendimiento. Sin embargo, se recomienda al menos:
   - 4 GB de memoria RAM.
   - 1 GB de espacio libre en disco.
   - Un procesador de doble núcleo o superior.
     Si tu equipo cumple con esto, podrás usar VS Code sin problemas para la mayoría de los proyectos.
2. **Una conexión a internet (para bajar un proyecto de prueba).**
   Aunque VS Code puede funcionar sin conexión una vez instalado, necesitarás internet al menos durante la configuración inicial para descargar:
   - El instalador desde la página oficial (https://code.visualstudio.com).
   - Las extensiones esenciales que activan soporte para distintos lenguajes.
   - Un proyecto de prueba que usarás para practicar (puede ser un repositorio de GitHub o un archivo comprimido con código ejemplo).
     Una vez tengas el entorno completo, podrás trabajar tanto online como offline.
3. **Reproducir “Hola Beats” para concentrarse y aprender más rápido.**
   El ambiente de aprendizaje influye directamente en la concentración y la retención de información. Escuchar música instrumental suave o con ritmo constante puede ayudarte a entrar en un estado de *flow* mental, ideal para aprender programación.
   Se recomienda reproducir la lista o el tema **“Hola Beats”**, disponible en Spotify, Apple Music o YouTube Music. Este tipo de música te ayuda a mantener el foco mientras exploras el entorno de VS Code, reduces distracciones y asimilas los comandos con mayor facilidad.
   Asegúrate de tener tus auriculares, ajustar el volumen a un nivel cómodo y prepararte mentalmente para sumergirte en la experiencia.

Conclusión de la lección

Al finalizar esta introducción, ya comprendes por qué aprender VS Code es una excelente inversión en tu desarrollo como programadora o creadora digital. Tienes las herramientas básicas y el entorno adecuado para empezar.
En la siguiente lección, aprenderás a **instalar VS Code paso a paso**, configurar las extensiones iniciales y realizar tu primera edición de código dentro del entorno.

II. Descarga e Instalación de VS Code

Una vez que conoces las razones por las que Visual Studio Code (VS Code) es una herramienta fundamental para cualquier programador o creador digital, el siguiente paso es aprender a **descargarlo, instalarlo correctamente y familiarizarte con su primera interfaz**.

Esta lección te guiará paso a paso desde el momento en que abres tu navegador hasta que ves la ventana principal del programa funcionando por primera vez. Aunque el proceso es muy sencillo, hacerlo con calma te garantizará que VS Code quede bien configurado desde el inicio.

A. Descarga

Antes de instalar VS Code, es necesario obtener el instalador desde su fuente oficial. No se recomienda descargarlo desde sitios no oficiales, ya que podrían contener versiones antiguas o incluso software modificado que comprometa la seguridad de tu equipo.

Sigue atentamente estos pasos:

1. **Ir al explorador (por ejemplo, Microsoft Edge).**
   En tu computador, abre tu navegador de internet preferido. Puedes usar **Microsoft Edge**, **Google Chrome**, **Mozilla Firefox** o cualquier otro navegador moderno.
   Lo importante es tener acceso a la web para buscar la página oficial de Visual Studio Code.

2. **Escribir “VS Code” en la barra de direcciones.**
   En la parte superior del navegador, donde aparece el espacio para escribir direcciones web, escribe:
   `VS Code`
   y presiona **Enter**.
   Esto abrirá los resultados de búsqueda con el sitio oficial y otros enlaces relacionados.

   También puedes escribir directamente la dirección:
   `https://code.visualstudio.com`
   para ir directamente al sitio oficial de Microsoft dedicado a VS Code.

3. **El enlace de búsqueda correcto es:**
   `code.visualstudio.com`
   Asegúrate de que el enlace que abras empiece exactamente con esa dirección.
   Si el navegador muestra un enlace que empieza diferente o contiene nombres sospechosos (por ejemplo, “vscode-downloads-now.com”), **no lo abras**.
   La dirección oficial siempre es **segura y legítima**, y en la barra de direcciones verás un pequeño candado indicando que la conexión está cifrada.

4. **Asegurarse de que aparezca “Download for Windows” o seleccionar la opción para el sistema operativo deseado.**
   Una vez dentro del sitio oficial, en la página principal verás un gran botón azul o morado con el texto:
   “Download for Windows”.
   Si usas otro sistema operativo, como macOS o Linux, debajo del botón principal verás opciones para cada uno.
   Elige la versión que corresponda a tu sistema.

- En **Windows**, suele descargarse un archivo con extensión `.exe`.
- En **macOS**, se descarga un archivo `.zip` o `.dmg`.
- En **Linux**, hay paquetes `.deb`, `.rpm` o `.tar.gz`.

5. **Hacer clic en "Download for Windows".**
   Al hacer clic, el sitio comenzará a descargar automáticamente el instalador.
   Dependiendo de tu conexión, esto puede tardar unos segundos o un par de minutos.
   Cuando la descarga termine, verás el archivo en la carpeta **Descargas** (Downloads) o en la parte inferior del navegador, lista para abrir.

B. Instalación en Windows

Ahora que ya tienes el instalador, es momento de ejecutar el proceso de instalación. Este procedimiento es muy rápido, pero es importante seguirlo con precisión para que VS Code quede completamente funcional.

1. **Abrir el archivo de instalación (Open file o buscar en Downloads).**
   Una vez terminada la descarga, puedes hacer clic directamente en el archivo desde tu navegador seleccionando **"Open file"** (Abrir archivo).
   Si ya cerraste el navegador, no pasa nada: abre tu carpeta **Descargas** y busca un archivo con nombre similar a:
   `VSCodeSetup-x64-x.x.x.exe`
   (donde "x.x.x" es el número de versión).
   Haz doble clic en el archivo para comenzar.

2. **Aceptar los términos y condiciones (primera opción).**
   Aparecerá la ventana del instalador con el acuerdo de licencia.
   Marca la casilla que dice **"I accept the agreement"** o **"Acepto el acuerdo"** y haz clic en **"Next"**.
   Este paso es necesario para continuar con la instalación.

3. **Seguir presionando "Siguiente".**
   El instalador te mostrará varias pantallas consecutivas:
   - Dónde se instalará el programa (puedes dejar la ruta por defecto).
   - Si quieres crear accesos directos en el menú de inicio o el escritorio.
   - Si deseas añadir opciones al menú contextual de Windows (como "Abrir con Code").
     En todos los casos, puedes dejar las opciones predeterminadas y simplemente presionar **"Next"** hasta llegar a la pantalla final.

4. **Finalizar con "Install".**
   En la última pantalla verás el botón **"Install"**.
   Haz clic sobre él y espera unos segundos mientras el programa copia todos los archivos necesarios.
   Durante este proceso, verás una barra de progreso que indica el avance de la instalación.
   Cuando llegue al final, aparecerá un mensaje de confirmación indicando que la instalación ha concluido con éxito.

5. **Al terminar, seleccionar “Launch Visual Studio Code” y presionar “Finish”.**
   Antes de cerrar el instalador, asegúrate de marcar la casilla **“Launch Visual Studio Code”** (Iniciar Visual Studio Code).
   Luego haz clic en **“Finish”**.
   Inmediatamente se abrirá la ventana principal del programa.
   ¡Felicidades! Acabas de instalar con éxito uno de los entornos de desarrollo más potentes del mundo.

C. Primera Vista

Una vez abierto por primera vez, VS Code te mostrará su **pantalla de bienvenida**, una interfaz limpia y moderna pensada para que comiences a trabajar de inmediato. A continuación, te explico qué verás y qué significa cada parte.

1. **Muestra una ventana de bienvenida (Get Started).**
   En el centro de la pantalla aparece una sección llamada **“Get Started”** o **“Comenzar”**.
   Aquí verás accesos rápidos a acciones iniciales como crear archivos, abrir proyectos o personalizar el entorno.
   También notarás una barra lateral izquierda con varios iconos:
   - Un documento (para explorar archivos).
   - Un icono de búsqueda (para buscar dentro del código).
   - Un símbolo de control de versiones (Git).
   - Un icono de depuración (para ejecutar y analizar código).
   - Un engranaje (para ajustes y extensiones).
     Esta barra será tu centro de control principal dentro del programa.
2. **Opciones: crear archivo, abrir archivo, abrir carpeta.**
   En la pantalla de bienvenida verás tres botones principales:
   - **New File (Nuevo archivo):** crea un nuevo documento vacío donde podrás escribir código.
   - **Open File (Abrir archivo):** te permite seleccionar un archivo existente en tu equipo.
   - **Open Folder (Abrir carpeta):** abre una carpeta completa, ideal para trabajar con proyectos que contienen varios archivos.
     La opción de abrir carpetas es especialmente útil para proyectos completos de programación.
3. **Mostrará proyectos trabajados anteriormente y tutoriales (aunque no se usarán).**
   Si ya habías usado VS Code antes, aparecerá una lista de tus proyectos recientes, lo que te permite reabrirlos fácilmente.
   Además, verás enlaces a tutoriales, guías y videos oficiales de Microsoft sobre cómo usar el programa.
   Aunque por ahora no los utilizaremos, estos recursos están disponibles para profundizar más adelante.

En este punto, lo importante es **explorar la interfaz, familiarizarte con los menús y entender la estructura general del editor**. No es necesario configurar nada aún; simplemente observa cómo está organizado todo.

Conclusión de la lección

Con esta lección, has aprendido a descargar e instalar correctamente VS Code y a reconocer su primera interfaz.
Ya tienes el entorno preparado para comenzar a **configurarlo, personalizarlo y escribir tu primer código**.

En la próxima lección descubrirás cómo ajustar su apariencia, cambiar el idioma, instalar extensiones útiles y dar tus primeros pasos en la edición de archivos.

III. Configuración Inicial (Idioma)

Una vez que Visual Studio Code está instalado y funcionando, el primer paso antes de empezar a programar es **ajustar el idioma de la interfaz**.
Por defecto, VS Code se instala en inglés, pero tiene la opción de cambiar fácilmente a otros idiomas mediante extensiones oficiales.

Aprender a manejar el idioma no solo mejora tu comodidad visual, sino que también te familiariza con los comandos más utilizados, que suelen encontrarse en inglés en la mayoría de tutoriales y documentación técnica.
En esta lección aprenderás:

- Cómo identificar el idioma por defecto.
- Cómo cambiarlo a español paso a paso.
- Cómo revertirlo a inglés si lo deseas más adelante.

A. Idioma por defecto (Inglés)

1. **Se ven menús en inglés (File, Edit, View, Go, Run, Terminal).**
   Al abrir Visual Studio Code por primera vez, notarás que su interfaz está completamente en inglés.
   En la parte superior encontrarás la **barra de menús**, con las siguientes opciones principales:
   - **File** → Archivo (para abrir, guardar o crear documentos).
   - **Edit** → Editar (copiar, pegar, buscar y reemplazar texto).
   - **View** → Ver (mostrar u ocultar paneles y barras).
   - **Go** → Ir (navegar entre archivos o líneas específicas).
   - **Run** → Ejecutar (iniciar la depuración o correr código).
   - **Terminal** → Terminal (abrir una consola integrada para escribir comandos).
   - **Help** → Ayuda (documentación, información del producto, versión, etc.).

   Además, los botones, mensajes emergentes y paneles también aparecen en inglés, como *New File*, *Open Folder*, *Settings*, *Extensions* o *Output*.

2. **Recomendación de aprenderlo en inglés, pero se explica cómo cambiarlo.**
   Aunque tener el entorno en inglés es útil para acostumbrarse a la terminología técnica (ya que la mayoría de recursos de programación están en ese idioma), es totalmente válido preferir usarlo en español mientras se aprende.

   VS Code permite cambiar el idioma fácilmente mediante una extensión oficial desarrollada por **Microsoft**, sin afectar el funcionamiento del programa.
   A continuación, verás el proceso completo para traducir la interfaz al español.

B. Cambio de idioma a Español

Cambiar el idioma en Visual Studio Code es un proceso muy sencillo y reversible. Solo necesitas acceder al sistema de **extensiones**, instalar el paquete de idioma español y reiniciar el programa.

1. **Ir al botón de extensiones.**
   En la barra lateral izquierda de VS Code, verás un ícono con cuatro pequeños cuadrados (uno separado del resto).
   Ese ícono representa el **Marketplace de Extensiones**.
   Haz clic en él para abrir la sección donde se buscan e instalan complementos.
   También puedes acceder rápidamente presionando:
   Ctrl + Shift + X

2. **Escribir "spanish" en la caja de búsqueda.**
   En la parte superior aparecerá una caja con el texto *Search Extensions in Marketplace* (Buscar extensiones en el Marketplace).
   Escribe la palabra:
   spanish
   Automáticamente se mostrarán varias opciones de extensiones relacionadas con el idioma español.

3. **Instalar la extensión de idioma español (creada por Microsoft, con 3 millones de descargas).**
   Busca entre los resultados la que se llama:
   "Spanish Language Pack for Visual Studio Code"
   Esta es la **extensión oficial** creada por **Microsoft**.
   Al lado del nombre verás que tiene millones de descargas (alrededor de 3 millones o más) y una alta valoración de usuarios.

   Haz clic en el botón azul que dice **Install** (Instalar).
   La instalación tardará unos segundos, y una vez completada, VS Code te sugerirá reiniciar el programa para aplicar los cambios.

4. **Reiniciar VS Code para aplicar el cambio.**
   Una vez instalada la extensión, aparecerá un mensaje en la esquina inferior derecha indicando que es necesario reiniciar.
   Haz clic en **Restart Now** o simplemente cierra el programa y vuelve a abrirlo.
   Al iniciar nuevamente, notarás que los menús, paneles y mensajes del sistema están completamente en español.
   Por ejemplo, ahora verás "Archivo", "Editar", "Ver", "Ejecutar" y "Terminal" en lugar de sus equivalentes en inglés.

5. **Método alternativo para cambiar el idioma: usar la Paleta de Comandos (Ctrl + Shift + P).**
   Si prefieres cambiar el idioma sin usar el panel de extensiones, también puedes hacerlo directamente con la **Paleta de Comandos**, una herramienta muy poderosa de VS Code.

   a. **Escribir "configure display".**
   Presiona **Ctrl + Shift + P** (o **Cmd + Shift + P** en macOS) para abrir la Paleta de Comandos.
   En la parte superior aparecerá una barra de búsqueda.

Escribe:
`configure display`
Aparecerá una opción llamada:
“Configure Display Language” (Configurar idioma de pantalla).
Selecciónala con **Enter**.

b. **Seleccionar idioma (English o Español).**
A continuación, se abrirá una lista de idiomas disponibles en tu instalación.
Si ya tienes el paquete de idioma español, verás la opción **“español”**.
Selecciónala y VS Code te pedirá reiniciar.
Una vez reiniciado, toda la interfaz cambiará al idioma elegido.

C. Revertir a Inglés

En caso de que más adelante quieras volver al idioma original (por ejemplo, para seguir tutoriales o acostumbrarte al inglés técnico), el proceso de reversión es igual de sencillo.

1. **Presionar Ctrl + Shift + P.**
   Abre nuevamente la **Paleta de Comandos** con la combinación:
   Ctrl + Shift + P
   o **Cmd + Shift + P** si usas macOS.
   Esta herramienta te permite ejecutar cualquier comando de configuración sin buscar en los menús.
2. **Seleccionar “Configurar idioma de pantalla”.**
   En la barra de búsqueda escribe:
   `configurar idioma`
   o en inglés:
   `configure display language`.
   Cuando aparezca la opción **“Configurar idioma de pantalla”**, selecciónala con **Enter**.
3. **Seleccionar “English”.**
   Verás una lista con los idiomas disponibles, incluyendo “English (en)” y “Español (es)”.
   Selecciona **English (en)**.
   VS Code te mostrará un aviso de que el cambio se aplicará después de reiniciar.
4. **Reiniciar VS Code.**
   Cierra el programa y vuelve a abrirlo.
   Verás que la interfaz vuelve al idioma original, con todos los menús y comandos en inglés.
   Esto no afectará tus archivos ni tus configuraciones personales, únicamente el idioma de visualización.

Conclusión de la lección

En esta lección aprendiste a:

- Reconocer el idioma por defecto de VS Code.
- Cambiar la interfaz al español mediante la extensión oficial de Microsoft.

- Usar la Paleta de Comandos para modificar el idioma sin entrar al Marketplace.
- Revertir el cambio fácilmente cuando lo necesites.

Dominar el sistema de idiomas es el primer paso para sentirte cómoda en el entorno de trabajo. Ahora que VS Code está configurado en el idioma que prefieras, estás lista para avanzar al siguiente nivel: **personalizar el entorno visual**, elegir temas de color, ajustar fuentes y explorar las extensiones que harán tu experiencia aún más fluida y atractiva.

IV. Componentes Principales de la Aplicación

Visual Studio Code (VS Code) es mucho más que un simple editor de texto: es un entorno de desarrollo potente, flexible y personalizable.
Sin embargo, para aprovechar su potencial, primero hay que **conocer a fondo su interfaz**.
Cada sección de la pantalla tiene una función específica, y entender su estructura te permitirá moverte con soltura, ahorrar tiempo y trabajar de manera más ordenada.

En esta lección exploraremos los **componentes principales de la aplicación**, esos elementos que verás cada vez que abras VS Code:
la **Barra de Actividades**, la **Barra de Estado** y el **Menú Superior**.

A. Barra de Actividades (Activity Bar)

1. **Se encuentra en la izquierda.**
   La **Barra de Actividades**, conocida en inglés como *Activity Bar*, se ubica verticalmente en el **lado izquierdo** de la ventana principal de VS Code.
   Es una columna de iconos que sirve para **acceder rápidamente a las funciones más importantes del programa**.
   Gracias a su diseño compacto, puedes tener a la vista tus herramientas principales sin ocupar demasiado espacio de pantalla.

   Esta barra puede personalizarse, mover algunos iconos o incluso ocultarse para maximizar el área de trabajo.
   Sin embargo, al principio es recomendable mantenerla visible para familiarizarte con todas sus secciones.

2. **Funcionalidades principales.**
   A continuación, veremos los iconos más importantes que aparecen por defecto en la Barra de Actividades y qué hace cada uno:

   a. **Explorador (archivos y carpetas).**
   Representado por un ícono de dos hojas o carpetas, el **Explorador** te permite **abrir, organizar y navegar entre archivos o proyectos**.
   Desde aquí puedes crear nuevos archivos, eliminar, mover, o arrastrar carpetas completas.
   Cuando abres una carpeta de proyecto, el Explorador muestra toda su estructura jerárquica, facilitando la gestión del código.

   b. **Buscar (texto dentro de archivos).**
   El ícono de una lupa abre el **buscador global**, que permite buscar palabras, frases o fragmentos de código en todo tu proyecto.
   También ofrece opciones avanzadas como búsqueda por mayúsculas/minúsculas, coincidencias exactas, uso de expresiones regulares y reemplazos masivos.
   Es una herramienta fundamental cuando trabajas con proyectos grandes y necesitas localizar funciones o variables específicas.

   c. **Control de Versiones (principalmente Git).**
   El icono con tres ramas conectadas corresponde al **control de versiones**, basado en el

sistema **Git**.
Desde aquí puedes realizar tareas como:

- Inicializar un repositorio local.
- Guardar cambios (*commits*).
- Sincronizar tu código con GitHub u otros servicios remotos.
- Revisar el historial de modificaciones.
- Comparar versiones anteriores de un archivo.
  Esta integración nativa convierte a VS Code en una herramienta perfecta para colaborar en proyectos compartidos.

d. **Debugger (se verá más adelante).**
El icono con un triángulo de "play" y un bicho o punto indica el **Depurador (Debugger)**.
Su función es ejecutar el código paso a paso, detectar errores, inspeccionar variables y analizar el flujo lógico del programa.
Es una herramienta avanzada que se estudia más adelante en el curso, una vez que empieces a trabajar con código ejecutable.

e. **Marketplace (para instalar extensiones).**
El icono con cuatro pequeños cuadrados (uno separado del grupo) abre el **Marketplace de Extensiones**.
Aquí puedes buscar, instalar, actualizar o eliminar extensiones que amplían las capacidades de VS Code.
Hay miles disponibles: temas visuales, autocompletado avanzado, soporte para nuevos lenguajes, conectores de bases de datos, entre muchas otras.
El Marketplace es el corazón de la personalización de VS Code y una de las razones por las que se considera un editor tan poderoso.

3. **Opción para Iniciar Sesión (sincronizar configuración en múltiples dispositivos).**
   En la parte superior de la Barra de Actividades encontrarás un pequeño ícono con una figura de usuario o círculo.
   Desde ahí puedes **iniciar sesión con una cuenta de Microsoft o GitHub**.
   Esto permite sincronizar automáticamente tus ajustes, temas, extensiones y atajos entre diferentes computadoras.
   Así, si usas VS Code en tu portátil y en un PC de escritorio, tendrás siempre el mismo entorno visual y las mismas configuraciones sin necesidad de reconfigurar nada manualmente.

   Esta sincronización es muy útil si trabajas en varios lugares o cambias de equipo con frecuencia.

4. **Engranaje (configurar preferencias y ajustes).**
   En la parte inferior de la Barra de Actividades se encuentra un ícono con forma de engranaje .
   Al hacer clic sobre él, se abre un menú con accesos rápidos a las principales configuraciones:

- **Settings (Configuración):** donde puedes cambiar fuentes, colores, tamaño de texto, comportamiento de archivos, etc.
- **Keyboard Shortcuts (Atajos de teclado):** para personalizar combinaciones de teclas.
- **Themes (Temas):** para cambiar el aspecto visual del editor.
- **Extensions (Extensiones):** para administrar complementos.
- **Command Palette (Paleta de Comandos):** una herramienta que permite ejecutar cualquier acción mediante texto.
  Este engranaje es, en esencia, la puerta de entrada a la **personalización completa del entorno**.

B. Barra de Estado (Status Bar)

1. **Barra inferior (inicialmente morada o celeste).**
   La **Barra de Estado**, o *Status Bar,* se encuentra en la parte **inferior** de la ventana de VS Code.
   Al iniciar el programa, normalmente tiene un color **morado** o **celeste**, dependiendo del tema visual elegido.
   Es un espacio informativo que cambia dinámicamente según el contexto del archivo o proyecto con el que trabajes.

   Aunque al principio puede parecer discreta, la Barra de Estado es muy útil: muestra información en tiempo real sobre lo que estás haciendo y permite acceder rápidamente a funciones del entorno.

2. **Tiene funcionalidades que se verán al editar archivos.**
   A medida que empieces a escribir código, la Barra de Estado mostrará detalles como:
   - El **lenguaje** del archivo actual (por ejemplo, Python, HTML, C#).
   - El **número de línea y columna** donde se encuentra el cursor.
   - El **estado del repositorio Git** (si hay cambios pendientes o conflictos).
   - El **modo de codificación de caracteres** (UTF-8, por ejemplo).
   - El **tipo de final de línea** (Windows `CRLF` o Unix `LF`).
   - Indicadores de **errores y advertencias** en el código.
     Además, algunos complementos añaden iconos propios a esta barra, como medidores de rendimiento, información del entorno virtual o estados del servidor.
     La Barra de Estado se convierte, con la práctica, en una guía constante sobre el estado actual de tu proyecto.

C. Menú Superior

1. **Opciones para personalizar la vista de VS Code.**
   En la parte superior de la ventana principal se encuentra el **Menú Superior**, una franja

horizontal con varias categorías desplegables.
Desde aquí puedes acceder a **todas las funciones internas del programa**, incluso aquellas que no están visibles en los iconos laterales.
Este menú te permite personalizar la vista, abrir paneles, configurar el diseño y ejecutar comandos directamente.

2. **Menú superior izquierdo con todas las opciones y acciones.**
   Las categorías principales que encontrarás son:
   - **File (Archivo):** crear, abrir, guardar, importar o cerrar archivos y proyectos.
   - **Edit (Editar):** copiar, pegar, buscar y reemplazar texto, deshacer y rehacer cambios.
   - **Selection (Selección):** modificar y manipular bloques de código o texto seleccionado.
   - **View (Ver):** mostrar u ocultar paneles, terminales, minimapas y vistas adicionales.
   - **Go (Ir):** navegar entre archivos, símbolos o líneas específicas.
   - **Run (Ejecutar):** iniciar y detener la depuración o ejecución de programas.
   - **Terminal:** abrir la terminal integrada, crear nuevas instancias o cambiar el tipo de consola.
   - **Help (Ayuda):** acceder a documentación oficial, versiones del programa y reportes de errores.

   Aunque con el tiempo muchos desarrolladores usan atajos de teclado para evitar abrir los menús, conocer su ubicación es fundamental al principio, ya que ofrece una visión completa de todas las funciones disponibles en VS Code.

Conclusión de la lección

Con esta lección has aprendido a identificar y comprender los tres componentes principales de Visual Studio Code:

- **La Barra de Actividades**, donde se concentran las herramientas esenciales.
- **La Barra de Estado**, que muestra información dinámica sobre tu trabajo.
- **El Menú Superior**, que centraliza todas las opciones y configuraciones del entorno.

Conocer su estructura es clave para navegar sin confusiones y sentirte cómoda dentro del programa. En la siguiente lección exploraremos cómo **personalizar la apariencia visual del entorno (temas, colores, fuentes y organización del espacio de trabajo)** para hacerlo más agradable y adaptado a tu estilo.

V. Uso del Explorador de Archivos (Explorer)

El **Explorador de Archivos** de VS Code (conocido como *Explorer*) es una de las herramientas más importantes del entorno. A través de él puedes navegar, crear, eliminar, renombrar, mover y organizar todos los archivos y carpetas que forman parte de tus proyectos. En esta lección aprenderás a utilizarlo paso a paso, comprendiendo cada una de sus funciones y herramientas disponibles.

A. Acceso

1. **Pinchar el primer botón en la Barra de Actividades.**
   En la parte izquierda de la ventana de VS Code se encuentra la *Barra de Actividades* con varios iconos. El primero, con el símbolo de una carpeta, es el acceso directo al **Explorador de Archivos**. Al hacer clic sobre él, se abrirá el panel lateral mostrando los archivos y carpetas del proyecto actual.
2. **Ocultar o mostrar el panel del explorador.**
   Si vuelves a pinchar el mismo icono, el panel se ocultará. Esto es útil para tener más espacio al escribir código. Puedes mostrarlo nuevamente con el mismo botón o utilizando el atajo de teclado **Ctrl+B**.

B. Apertura de Carpeta

Antes de comenzar a trabajar, VS Code necesita saber en qué carpeta estarán tus archivos del proyecto.

1. **Hacer clic en “Open Folder”.**
   Si es la primera vez que abres el programa, aparecerá un mensaje en el panel principal invitándote a abrir una carpeta. Haz clic en el botón **“Open Folder”**.
2. **Crear una nueva carpeta (por ejemplo, `tuto-vscode`).**
   Si aún no tienes un proyecto, puedes crear una nueva carpeta. Elige una ubicación fácil de recordar, como el Escritorio o la carpeta Documentos.
   - Escribe el nombre `tuto-vscode`.
   - Presiona **Enter** para crearla.
3. **Seleccionar la carpeta creada.**
   Haz clic sobre la carpeta `tuto-vscode` y luego en **“Seleccionar carpeta”** (o *Select Folder*). VS Code abrirá esa carpeta como el entorno de trabajo.

C. Botones del Explorer

En la parte superior del panel del *Explorer* hay varios botones pequeños con iconos. Cada uno cumple una función útil en la gestión de archivos.

1. **Crear archivos.**
   El primer icono (una hoja con un “+”) permite crear nuevos archivos. Al hacer clic,

aparecerá un cuadro para escribir el nombre del archivo, por ejemplo:
`index.md`

2. **Crear carpetas.**
   El segundo icono (una carpeta con un "+") sirve para crear nuevas carpetas dentro del proyecto. Esto te permite organizar tus archivos por tipo o funcionalidad.
3. **Refrescar.**
   El tercer icono (dos flechas circulares) actualiza el contenido mostrado en el *Explorer*. Es muy útil cuando agregas o borras archivos desde fuera de VS Code (por ejemplo, desde el Explorador de Windows), ya que recarga la vista.
4. **Colapsar todas las carpetas.**
   El cuarto icono (dos carpetas pequeñas con una flecha hacia arriba) cierra todas las carpetas que estén abiertas, dejando la vista del *Explorer* más ordenada y compacta.

D. Creación y Edición de Archivos

Ahora que tienes abierta una carpeta, puedes empezar a crear y editar archivos dentro de ella.

1. **Crear un archivo (por ejemplo, `index.md`).**
   Haz clic en el botón *Nuevo archivo* e introduce el nombre `index.md`.
   El archivo se abrirá automáticamente en el editor principal.
2. **Escribir texto.**
   En el área de edición, escribe una línea de texto, por ejemplo:

   ```
   hola mundo
   ```
3. **Guardar cambios.**
   Para guardar el archivo, ve al menú **File → Save** o usa el atajo **Ctrl+S**.
   Cada vez que guardas, el contenido del archivo se actualiza en disco.
4. **Indicador de cambios.**
   - Si aparece un **círculo** en la pestaña del archivo, significa que hay cambios **no guardados**.
   - Si aparece una **"x"**, significa que el archivo está guardado y puede cerrarse sin perder información.
5. **Crear otro archivo (por ejemplo, `readme.md`).**
   Repite el proceso creando un nuevo archivo llamado `readme.md`.
   Puedes escribir una breve descripción de tu proyecto.
6. **Uso de pestañas.**
   Cada archivo abierto aparece en la parte superior como una pestaña. Puedes tener varios archivos abiertos al mismo tiempo y cambiar entre ellos haciendo clic en sus nombres.
7. **Cerrar pestañas.**
   Para cerrar una pestaña, haz clic en la **"x"** del costado derecho del nombre del archivo. Si hay cambios sin guardar, VS Code te preguntará si deseas hacerlo antes de cerrar.

E. Organización de Archivos

A medida que tu proyecto crece, es necesario mantener todo bien estructurado.

1. **Crear una carpeta (por ejemplo, `src`).**
   Haz clic en el botón *Nueva carpeta* y escribe el nombre `src`. Esta carpeta servirá para guardar el código fuente.
2. **Usos de la carpeta `src`.**
   Es una convención común en desarrollo de software almacenar todo el código del proyecto dentro de una carpeta llamada `src` (*source* o fuente). Esto mantiene separadas las configuraciones y documentación de los archivos de código.
3. **Mover archivos.**
   Arrastra los archivos `index.md` y `readme.md` hacia la carpeta `src`. Verás cómo se mueven dentro de ella y la estructura del proyecto se actualiza en tiempo real.
4. **Colapsar u ocultar contenido.**
   Puedes ocultar el contenido de una carpeta haciendo clic sobre el nombre o el icono de la flecha a su izquierda. Esto es muy útil cuando hay muchos archivos y quieres mantener el *Explorer* limpio.

F. Manipulación de Texto

El editor de texto de VS Code es muy poderoso y rápido. Veamos algunas funciones básicas de edición.

1. **Pegar texto copiado.**
   Copia un fragmento de texto, por ejemplo un párrafo de *Lorem Ipsum*, y pégalo en `index.md` usando **Ctrl+V**.
2. **Visualización del texto.**
   Si el texto es largo, VS Code lo mostrará en una sola línea. A la izquierda podrás ver el número “1”, lo que indica que todo el contenido está en una sola línea. Esto se debe al **ajuste de línea** (*Word Wrap*) que puede activarse o desactivarse desde el menú **View → Word Wrap**.
3. **Separar texto en varias líneas.**
   Coloca el cursor donde desees hacer un salto de línea y presiona **Enter**. El texto se dividirá, y los números de línea cambiarán automáticamente. Luego guarda con **Ctrl+S** para confirmar los cambios.

Conclusión

El **Explorador de Archivos** es el corazón del flujo de trabajo en VS Code. Desde él puedes abrir, organizar y manipular todos los recursos de tus proyectos. Dominarlo es el primer paso para convertir VS Code en tu centro de desarrollo, ya sea que programes en Python, JavaScript, C#, o simplemente trabajes con documentos Markdown.

Una buena práctica es mantener tu proyecto siempre ordenado: carpetas claras, nombres descriptivos y archivos bien clasificados. Esto no solo mejora la productividad, sino que también hace que tu experiencia en VS Code sea mucho más fluida y agradable.

VI. Funcionalidades de la Barra de Estado (Status Bar)

La **Barra de Estado** (*Status Bar*) es una de las secciones más útiles y a menudo menos comprendidas de Visual Studio Code.
Se ubica en la parte **inferior de la ventana**, justo debajo del área de edición, y proporciona **información en tiempo real** sobre el estado actual del editor, el archivo abierto, la configuración del entorno y el lenguaje activo.

Esta barra cambia dinámicamente según el contexto: muestra distintos indicadores dependiendo del tipo de archivo que se edite, las extensiones instaladas o incluso el sistema de control de versiones que se esté usando.
A continuación, se explican detalladamente sus componentes principales y cómo sacarles el máximo provecho.

A. Indicadores Izquierdos

1. **Indicadores de errores y advertencias.**
   En el extremo izquierdo de la barra de estado pueden aparecer **pequeños íconos amarillos o rojos**:

   - Un **triángulo amarillo** indica advertencias (*warnings*).
   - Un **círculo rojo con una "x"** indica errores (*errors*).

   Estos indicadores son generados por el motor de análisis del lenguaje que esté activo (por ejemplo, TypeScript, Python o C#).

   Si tu archivo contiene errores de sintaxis, o advertencias sobre prácticas no recomendadas, estos íconos te lo informan sin necesidad de ejecutar el programa.

   > **Consejo:** Puedes hacer clic directamente sobre el indicador para abrir la pestaña de *Problemas (Problems)*, donde se muestran los errores y advertencias con detalles, línea afectada y sugerencias de solución.

B. Posición del Cursor

1. **Visualización de la posición.**
   En la parte derecha de la barra de estado verás un indicador con formato similar a:

   ```
   Ln 11, Col 1
   ```

   Esto significa que el cursor se encuentra en la **línea 11** y la **columna 1** del archivo.

2. **Utilidad.**
   Este indicador es muy útil cuando otros desarrolladores te dicen "revisa la línea 87" o "añade un punto en la columna 5".
   También es práctico al depurar código, buscar errores o seguir instrucciones en tutoriales técnicos.

3. **Navegar a una posición específica.**
   Si haces clic sobre el indicador (donde dice *Ln X, Col Y*), aparecerá una pequeña ventana en

la parte superior del editor.
Allí puedes escribir la posición exacta a la que deseas moverte, por ejemplo:

```
5,10
```

Luego presiona **Enter** y el cursor saltará directamente a la **línea 5, columna 10**.

> **Atajo alternativo:** también puedes usar **Ctrl+G** para abrir la ventana de navegación rápida a línea específica.

C. Configuración de Identación (Indentation)

1. **Visualización del tipo de indentación.**
   Justo al lado de los indicadores de posición, aparece un texto que puede decir algo como:

   ```
   Spaces: 4
   ```

   o bien:

   ```
   Tab Size: 4
   ```

   Esto indica cómo está configurada la **indentación** del archivo actual, es decir, la manera en que se sangran las líneas del código.

2. **Opciones comunes.**

   - **4 espacios:** es el estándar más usado en Python, HTML y muchos otros lenguajes.
   - **2 espacios:** común en proyectos de JavaScript, Vue o JSON.
   - **Tabulación (Tab):** usa el carácter especial de tabulación en lugar de espacios.

   > La elección depende del estilo del equipo o del lenguaje de programación, pero es fundamental mantener consistencia dentro del mismo proyecto.

3. **Configuración de la indentación.**
   Haz clic en el texto que muestra la indentación (por ejemplo, “Spaces: 4”) y se desplegará un menú con varias opciones:

   - **Convert Indentation to Spaces** (*Convertir sangría a espacios*).
   - **Convert Indentation to Tabs** (*Convertir sangría a tabulaciones*).
   - **Indent Using Spaces** (*Indentar utilizando espacios*).
   - **Indent Using Tabs** (*Indentar utilizando tabs*).
   - **Detect Indentation from File** (*Detectar la indentación del archivo*).

   Estas opciones permiten adaptar el formato del archivo actual o del proyecto entero con solo un clic.

   > **Recomendación:** define un valor fijo (por ejemplo, 4 espacios) en tus ajustes globales para mantener una apariencia coherente entre todos tus archivos.

D. Codificación y Final de Línea

En el extremo derecho de la barra de estado encontrarás dos indicadores fundamentales: **la codificación** del archivo y **el tipo de fin de línea**.

1. Codificación del archivo

- El formato más común es:

  `UTF-8`

  Este estándar es el más popular en todo internet porque **soporta la mayoría de idiomas y símbolos** (incluyendo tildes, eñes, caracteres chinos, etc.).

- **Concepto de codificación:**
  Cada carácter que ves en pantalla tiene un valor numérico. Por ejemplo, la letra `'A'` equivale al número **65** en el sistema ASCII.
  Las codificaciones como **UTF-8** amplían ese concepto para incluir millones de caracteres de diferentes alfabetos y símbolos.

- Si un archivo no está correctamente codificado, pueden aparecer símbolos raros o signos de interrogación donde deberían ir letras acentuadas.

- **Cambio de codificación:**
  Al hacer clic sobre “UTF-8”, VS Code ofrece opciones como “Reopen with Encoding” o “Save with Encoding”.
  Esto te permite reabrir o guardar el archivo con una codificación distinta si fuera necesario (por ejemplo, ISO-8859-1 o UTF-16).

---

2. Final de línea (EOL: End of Line)

Las computadoras representan los saltos de línea de forma distinta según el sistema operativo.
VS Code muestra esta información en la barra de estado con etiquetas como:

- **Windows:** `CRLF` (*Carriage Return + Line Feed*)
- **macOS y Linux:** `LF` (*Line Feed*)
- **macOS clásico:** `CR` (*Carriage Return*)

Importancia práctica:

Cuando trabajas en equipo con desarrolladores que usan distintos sistemas operativos, las diferencias en los finales de línea pueden causar conflictos al compartir código o archivos de texto.

Por ejemplo, un archivo guardado con `CRLF` en Windows puede generar errores de formato al abrirse en Linux si el sistema no los interpreta correctamente.

Soluciones y buenas prácticas:

- Usa un sistema de control de versiones como **Git**, que puede normalizar automáticamente los finales de línea.
- Establece en tu configuración de VS Code una preferencia fija (por ejemplo, siempre usar `LF`).

Puedes hacerlo desde **Configuración → Text Editor → Files → EOL**.

E. Tipo de Archivo / Lenguaje

1. **Visualización del lenguaje actual.**
   En el extremo derecho de la barra de estado aparece el nombre del lenguaje o tipo de archivo actual, por ejemplo:

   `Markdown`

   o

   `Python`

   VS Code detecta automáticamente el lenguaje según la extensión del archivo (`.md`, `.py`, `.js`, etc.) y activa el resaltado de sintaxis correspondiente.

2. **Cambio manual del lenguaje.**
   Si el editor no detecta el lenguaje correctamente o si deseas probar otro, puedes hacer clic en el nombre del lenguaje y aparecerá un menú con cientos de opciones disponibles.
   Por ejemplo:

   - Markdown
   - JavaScript
   - HTML
   - JSON
   - C++
   - CoffeeScript
   - YAML

   Incluso sin instalar extensiones, VS Code soporta decenas de lenguajes por defecto.
   Si necesitas soporte adicional (como autocompletado o depuración), puedes instalar extensiones desde el *Marketplace*.

Conclusión

La **Barra de Estado** de Visual Studio Code es mucho más que una línea decorativa:
es una fuente constante de información técnica sobre tu entorno de trabajo, tu archivo actual y la configuración activa del editor.

Dominarla te permitirá:

- Detectar errores antes de ejecutar el código.
- Navegar con precisión dentro de tus archivos.
- Mantener un formato de código coherente.
- Evitar problemas de codificación y compatibilidad entre sistemas.

- Cambiar rápidamente de lenguaje o modo de edición.

Comprender la *Status Bar* es dar un paso firme hacia un uso profesional de VS Code, optimizando tu flujo de trabajo y mejorando la calidad de tus proyectos.

VII. Búsquedas y Reemplazos

Una de las herramientas más potentes de Visual Studio Code es su **motor de búsqueda y reemplazo**, que permite localizar texto, fragmentos de código, variables o etiquetas en todo un proyecto de forma instantánea.
Esta función es indispensable para proyectos grandes con múltiples archivos, ya que evita recorrer manualmente línea por línea.

En esta lección aprenderás cómo usar la búsqueda general, sus filtros, y cómo realizar reemplazos masivos con seguridad y precisión.

A. Acceso

1. **Pinchar el icono de la lupa en la Barra de Actividades.**
   En la parte izquierda de la interfaz de VS Code, encontrarás la **Barra de Actividades**, donde se agrupan las herramientas principales del entorno.
   El **tercer icono**, con forma de **lupa**, corresponde a la función de **búsqueda** (*Search*).
   Al hacer clic sobre él, se abrirá el panel de búsqueda en la parte lateral izquierda del editor.

   > *Atajo rápido:* también puedes acceder al panel de búsqueda usando **Ctrl+Shift+F** en Windows o **Cmd+Shift+F** en macOS.

   El panel muestra una caja de texto superior (para introducir lo que se desea buscar) y varias opciones debajo para refinar la búsqueda.

B. Búsqueda General

La **búsqueda general** de VS Code no se limita al archivo abierto, sino que explora **toda la carpeta del proyecto** actualmente activa.

1. **Realizar una búsqueda personalizada.**
   En la caja superior del panel de búsqueda, escribe una palabra o fragmento que quieras localizar.
   Por ejemplo:

   ```
   but
   ```

   VS Code buscará esta cadena dentro de todos los archivos del proyecto.

2. **Visualización de resultados.**
   Una vez completada la búsqueda, verás una lista de coincidencias con el siguiente formato:

   ```
   readme.md — 2 results
   ```

   Debajo se mostrarán las líneas específicas donde aparece el texto buscado, con la coincidencia resaltada en color.

   Cada resultado indica:

   - **El nombre del archivo** donde se encuentra.
   - **El número de línea.**

- **Un extracto del texto con la palabra resaltada.**

Si haces clic sobre una de las coincidencias, VS Code abrirá automáticamente el archivo correspondiente y te llevará exactamente a la línea donde está el resultado.

> **Consejo:** Si tienes un proyecto muy grande, puedes filtrar las búsquedas por tipo de archivo o por ruta usando los cuadros “Include files” y “Exclude files” en la parte inferior del panel.

C. Opciones de Búsqueda

Para afinar la búsqueda y obtener resultados más precisos, VS Code incluye **filtros y modificadores** accesibles a través de pequeños iconos ubicados junto a la caja de texto principal.

1. **Match Case (A/a).**
   Este botón activa o desactiva la **sensibilidad a mayúsculas y minúsculas**.
   - Si está **activado**, buscar `"Hello"` no mostrará resultados que digan `"hello"`.
   - Si está **desactivado**, se mostrarán ambos.

   Ejemplo:
   Buscar **“Pain”** devolverá resultados diferentes a **“pain”** si el modo *Match Case* está activo.

2. **Whole Word (ab).**
   Esta opción permite buscar **palabras completas**.
   Cuando está activada, VS Code solo mostrará coincidencias que sean exactamente iguales a la palabra buscada, ignorando aquellas donde aparezca como parte de otra palabra.

   Ejemplo práctico:
   Si buscas **“Pain”** con *Whole Word* activado, **no** se mostrará la palabra *“Paintful”*.
   Si está desactivado, *“Paintful”* y *“Pain”* se considerarán coincidencias válidas.

   Esto resulta muy útil al editar código, especialmente al buscar nombres de variables o funciones que podrían formar parte de otros identificadores.

D. Reemplazo de Texto

Además de buscar, VS Code te permite **reemplazar texto** de forma controlada o masiva.
Esta función es esencial para tareas de mantenimiento, refactorización de código o corrección de errores globales.

1. **Activar la caja de reemplazo.**
   Debajo del campo de búsqueda verás otra caja con el texto *“Replace”* o *“Reemplazar”*.
   Si no aparece, puedes mostrarla haciendo clic en la pequeña **flecha hacia abajo** ubicada a la derecha del campo principal de búsqueda.

2. **Indicar el texto de reemplazo.**
   En la caja *Replace*, escribe el texto que sustituirá a las coincidencias encontradas.
   Por ejemplo, si buscaste **“Pain”** y quieres cambiarlo por **“nice”**, escribe:

`nice`

3. **Ejecutar el reemplazo.**
   Junto a las cajas de búsqueda y reemplazo hay varios botones:
   - **Replace** (flecha circular): reemplaza **una única coincidencia** seleccionada.
   - **Replace All** (dos flechas circulares): reemplaza **todas las ocurrencias** en todos los archivos.

   Al hacer clic en **Replace All**, VS Code mostrará un resumen de los cambios que se aplicarán. Puedes confirmar antes de ejecutar el reemplazo completo.

   > ⚠ **Advertencia:** usa “Replace All” con precaución.
   > Un reemplazo global mal definido puede modificar código o texto en lugares no deseados.
   > Se recomienda realizar una búsqueda previa sin reemplazo para verificar todas las coincidencias antes de confirmar.

Ejemplo Completo

Supongamos que tienes un proyecto con varios archivos Markdown (`readme.md`, `index.md`, `notas.md`) y quieres cambiar la palabra **“Pain”** por **“nice”** en todo el proyecto:

1. Abre el panel de búsqueda con **Ctrl+Shift+F**.
2. Escribe `"Pain"` en el cuadro superior.
3. Verifica los resultados en los archivos listados.
4. En el campo *Replace*, escribe `"nice"`.
5. Haz clic en **Replace All**.
6. VS Code te mostrará los cambios y actualizará todos los archivos en segundos.

Ahora todos los textos donde antes decía “Pain” habrán sido reemplazados por “nice”.

Conclusión

El sistema de **búsqueda y reemplazo** de Visual Studio Code es una herramienta profesional que combina **velocidad**, **precisión** y **seguridad**.
Dominarla te permitirá trabajar con proyectos de gran tamaño, modificar patrones de texto complejos y mantener tu código limpio de redundancias o errores tipográficos.

Resumen de ventajas:

- Busca y reemplaza texto en miles de archivos al instante.
- Distingue entre mayúsculas/minúsculas y palabras exactas.
- Permite previsualizar y confirmar los cambios antes de aplicarlos.
- Facilita la edición colaborativa en proyectos grandes.

Con práctica, esta herramienta se convierte en una extensión natural de tu flujo de trabajo dentro de VS Code, haciendo que editar, corregir o mejorar tu código sea mucho más rápido y eficiente.

VIII. Control de Versiones (Git)

El **control de versiones** es una de las herramientas más importantes en el desarrollo moderno. Permite llevar un registro de los cambios realizados en los archivos, colaborar con otros desarrolladores y restaurar versiones anteriores del proyecto en caso de errores.
Visual Studio Code (VS Code) tiene una integración directa con **Git**, el sistema de control de versiones más popular del mundo, utilizado por millones de desarrolladores y empresas para gestionar código fuente.

A. Instalación de Git

1. **Detección automática en VS Code:**
   Al abrir un proyecto en VS Code, si el programa detecta que Git no está instalado, mostrará un botón o una notificación indicando que se necesita instalarlo.
   En este caso, aparecerá la opción **"Download Git"** o **"Descargar Git"**.
2. **Descarga e instalación:**
   - Al hacer clic en el botón, se abrirá la página oficial de Git: https://git-scm.com/.
   - Descarga el instalador correspondiente a tu sistema operativo (Windows, macOS o Linux).
   - En Windows, el proceso de instalación es muy sencillo: basta con seguir la secuencia de botones **“Next” → “Next” → “Install”**.
   - Acepta las configuraciones por defecto a menos que sepas lo que estás modificando.
3. **Recargar VS Code:**
   Una vez completada la instalación, vuelve a abrir Visual Studio Code. Si estaba abierto, utiliza la opción **Reload** o **Recargar ventana** para que VS Code reconozca Git correctamente.

B. Inicialización del Repositorio

Una vez instalado Git, puedes inicializar tu primer repositorio directamente desde VS Code.

1. **Iniciar el repositorio:**
   - Abre la pestaña de **Control de versiones** haciendo clic en el ícono de la ramita con un punto (tercer botón de la barra de actividades).
   - Si tu proyecto no tiene un repositorio inicializado, verás la opción **“Initialize Repository”** o **“Iniciar repositorio”**.
   - Haz clic en ella y Git creará una carpeta oculta llamada `.git`, que almacenará todo el historial de versiones.
2. **Componentes principales de la vista de control de versiones:**
   - **Zona de cambios (Changes):** muestra los archivos modificados que aún no han sido guardados en el historial.

- **Zona de preparación (Staged Changes):** contiene los archivos que están listos para ser guardados (commit).
- **Caja de mensaje de commit:** aquí se escribe una breve descripción de los cambios realizados.
- **Botón de Commit:** al presionarlo, se registran los cambios con el mensaje indicado.

3. **Archivos cambiados:**
   Al iniciar un repositorio en un proyecto nuevo, todos los archivos aparecerán como nuevos. Puedes agregar los que quieras al "staging" usando el ícono de **“+”** junto a cada nombre de archivo.

C. Configuración de Usuario

Antes de poder hacer un commit por primera vez, Git necesita saber quién eres, es decir, tu nombre y tu dirección de correo electrónico.

1. **Error de configuración:**
   Si intentas hacer un commit sin configurar estos datos, VS Code mostrará un error indicando que faltan los valores `user.name` y `user.email`.
2. **Solución mediante Git Bash:**
   - En Windows, abre el programa **Git Bash** (puedes buscarlo desde el menú de inicio escribiendo “git”).
   - Luego, escribe los siguientes comandos, reemplazando los valores entre comillas con tu nombre y tu correo electrónico:

     ```
     git config --global user.name "Tu Nombre"
     ```

     ```
     git config --global user.email "tuemail@ejemplo.com"
     ```
3. **Limpiar la terminal:**
   Si la terminal está llena de texto y deseas limpiarla, puedes ejecutar el comando:

   ```
   clear
   ```

   Esto dejará la pantalla vacía para continuar trabajando cómodamente.

D. Conexión con GitHub y Publicación

GitHub es una plataforma en línea que permite almacenar tus repositorios en la nube y colaborar con otras personas. VS Code permite conectarse directamente a GitHub.

1. **Iniciar sesión:**
   - Haz clic en el menú **Branch** (Rama) y luego selecciona **“Sign in to GitHub”**.
   - Si es la primera vez, VS Code te pedirá iniciar sesión mediante el navegador.
   - Acepta los permisos y autoriza el **Git Credential Manager** para que VS Code recuerde tus credenciales de manera segura.

2. **Publicar el proyecto:**
   Una vez autenticado, VS Code mostrará la opción **“Publish to GitHub”** o **“Publicar en GitHub”**.
   Podrás elegir si deseas que el repositorio sea **público** (visible para todos) o **privado** (solo tú o tu equipo pueden acceder).
3. **Visualizar el repositorio en GitHub:**
   Cuando se complete la publicación, aparecerá un enlace que dice **“Open on GitHub”**.
   Al hacer clic, se abrirá tu navegador mostrando el repositorio en tu cuenta de GitHub.

E. Manejo de Ramas (Branches)

Las ramas son una característica esencial de Git que permiten trabajar en distintas versiones del mismo proyecto de manera paralela, sin afectar la versión principal (llamada **Master** o **Main**).

1. **Crear una nueva rama:**
   - Puedes hacerlo desde los tres puntos (…) en la parte superior del panel de control de versiones → **Branch** → **Create Branch**.
   - También puedes hacer clic en el nombre de la rama actual (por defecto, “master”) en la barra de estado inferior y elegir **“Create new branch”**.
   - Escribe un nombre descriptivo, por ejemplo: `nueva-rama`.
2. **Publicar la rama:**
   Una vez creada, selecciona la opción **“Publish branch”** para enviar la rama al repositorio remoto de GitHub.
3. **Hacer cambios en la nueva rama:**
   Edita un archivo (por ejemplo, `Index.md`), guarda los cambios y agrégalos al staging.
   Escribe un mensaje en la caja de commit, como “nuevos cambios”, y confírmalo.
4. **Sincronizar cambios:**
   Usa la opción **“Sync Changes”**, que ejecuta internamente las acciones **Pull** (traer cambios del servidor) y **Push** (enviar tus cambios).
5. **Cambiar de rama:**
   Puedes volver a la rama principal seleccionando **Master** desde el mismo menú inferior o desde el panel de control de versiones.
6. **Fusionar ramas (Merge):**
   - Para combinar los cambios de una rama secundaria con la principal, selecciona los tres puntos → **Branch** → **Merge Branch**.
   - Elige la rama que deseas fusionar (por ejemplo, `nueva-rama`).
   - Git incorporará los cambios en la rama principal.
7. **Eliminar una rama:**
   - Cuando la rama ya no sea necesaria, puedes eliminarla desde el mismo menú: **Branch** → **Delete Branch**.

- Selecciona la rama que deseas borrar para mantener tu repositorio ordenado.

Conclusión

El uso de **Git** en VS Code ofrece una manera integrada, visual y potente de gestionar proyectos. Desde la creación de repositorios y commits hasta la sincronización con GitHub y el manejo de ramas, todo se puede realizar sin salir del editor.
Dominar estas herramientas no solo mejora tu flujo de trabajo, sino que también te prepara para colaborar en entornos profesionales, donde el control de versiones es una práctica esencial.

IX. Depuración (Debugger)

La **depuración (debugging)** es el proceso de analizar un programa mientras se ejecuta para encontrar y corregir errores. Visual Studio Code (VS Code) cuenta con un **depurador integrado** que permite ejecutar paso a paso el código, observar el valor de las variables, entender la lógica de ejecución y descubrir por qué algo no funciona como se esperaba.
Esta herramienta es esencial para todo programador, ya que permite **detectar fallos con precisión** y **comprender el comportamiento interno** de las aplicaciones.

A. Requisitos y Obtención del Proyecto

1. **Conocimientos previos:**
   Para aprovechar al máximo esta sección, se requiere tener nociones básicas de programación —por ejemplo, saber qué son las variables, funciones y estructuras condicionales.
2. **Descargar o clonar el proyecto de prueba:**
   En este ejemplo, se utilizará un pequeño proyecto de JavaScript para practicar la depuración. Este proyecto puede descargarse manualmente desde un enlace o **clonarse** directamente desde GitHub utilizando las herramientas de VS Code.
3. **Clonar el proyecto desde GitHub:**
   - Abre la pestaña de **Control de versiones** (icono de ramita con un punto en la barra de actividades).
   - Haz clic en la opción **“Clone Repository”** o **“Clonar repositorio”**.
   - En la barra que aparece, escribe el nombre del repositorio:

     ```
     nschorm proyecto-js
     ```
   - VS Code buscará el repositorio automáticamente.
   - Selecciona una carpeta local donde se descargará el proyecto (por ejemplo, en tu carpeta “Documentos”).
4. **Abrir la carpeta clonada:**
   Una vez descargado, abre el proyecto desde el menú superior:
   File → Open Folder (Archivo → Abrir carpeta) y selecciona la carpeta `proyecto-js`.
   VS Code cargará los archivos del proyecto para trabajar en ellos.

B. Inicio del Debugger

1. **Preparación:**
   Cierra la pestaña inicial **“Get Started”** si está abierta para tener una vista más limpia del entorno.
   Luego, abre el archivo principal del proyecto, normalmente **`index.html`**, ya que será la puerta de entrada de la aplicación web.
2. **Abrir la vista de depuración:**
   Haz clic en el icono del **insecto** (Debug) en la barra de actividades de la izquierda.

Esta es la sección dedicada a la depuración. Desde aquí se pueden configurar los entornos de ejecución, los puntos de interrupción y los controles de ejecución.

3. **Configuración personalizada (opcional):**
   En algunos casos se puede crear un archivo `launch.json` para personalizar el comportamiento del depurador (por ejemplo, indicar parámetros específicos, rutas o navegadores).
   En esta lección **no será necesario**, ya que VS Code ofrece una opción automática para ejecutar y depurar.
4. **Ejecutar el programa:**
   - Haz clic en el botón **"Run and Debug"**.
   - El sistema te pedirá seleccionar el entorno. Elige **Chrome** o **Edge** (cualquiera que tengas instalado).
   - VS Code abrirá automáticamente una nueva ventana del navegador ejecutando la aplicación web de prueba.
5. **Visualizar la aplicación:**
   En el navegador aparecerá una página simple titulada **"Web de prueba"**, con un botón que dice **"cliquéame"**.
   Esta será la interfaz del proyecto que depuraremos.

C. Uso de Breakpoints

1. **Propósito:**
   Los **breakpoints (puntos de interrupción)** permiten detener la ejecución del programa en una línea específica.
   Esto da tiempo al desarrollador para examinar el estado de las variables, los valores, y el flujo del programa antes de que continúe.
2. **Cómo agregar un breakpoint:**
   - Abre el archivo **`index.js`** (el script asociado al HTML).
   - Haz clic en el área gris a la izquierda del número de línea donde deseas detener la ejecución (por ejemplo, en la línea 2).
   - Aparecerá un **círculo rojo** indicando que la línea tiene un breakpoint activo.
3. **Requisito importante:**
   Antes de iniciar la depuración, asegúrate de que el archivo **`index.html`** esté activo o visible en la ventana principal.
   Esto garantiza que el depurador pueda enlazar correctamente el código del navegador con el código fuente en VS Code.

D. Componentes de la Interfaz del Debugger

Cuando el depurador alcanza un breakpoint, se abrirá la **vista de depuración**, donde podrás analizar los detalles del programa en tiempo real.

1. **Panel de Variables:**
   Muestra el valor actual de todas las variables del programa.
   - **Variables locales:** son las que se encuentran dentro de la función que se está ejecutando.
     Ejemplo: `b`, `myString`.
   - **Variables de script:** son las que pertenecen al archivo actual o módulo.
     Ejemplo: `clickeado`, `FN`.
   - **Variables globales:** son las que se definen fuera de cualquier función o importadas desde otros scripts.
2. **Indicador de línea activa:**
   La línea donde se detuvo la ejecución se resalta en **amarillo**.
   Esto muestra el punto exacto donde el código ha sido pausado y donde puede analizarse su comportamiento.
3. **Panel de Call Stack (pila de llamadas):**
   Indica qué funciones se han llamado antes de llegar al punto actual.
   Es muy útil para rastrear el flujo de ejecución de una aplicación más compleja.
4. **Panel de Watch (observadores):**
   Permite escribir nombres de variables para vigilarlas constantemente, incluso cuando cambian de valor durante la ejecución.

E. Controles de Depuración

Durante la ejecución, los controles principales del depurador aparecen en la parte superior del editor.
Estos botones permiten avanzar, retroceder o detener el flujo de ejecución.

1. **Continuar / Play:**
   Reanuda la ejecución del programa hasta el siguiente breakpoint o hasta que termine.
2. **Step Over (F10):**
   Ejecuta la línea actual y pasa a la siguiente.
   Si la línea contiene una llamada a una función, **no entra dentro de ella**, sino que la ejecuta completa y continúa.
3. **Step Into:**
   Entra dentro de la función llamada en la línea actual.
   Es útil para analizar el comportamiento interno de una función.
4. **Step Out:**
   Sale del nivel actual de depuración y regresa al nivel anterior (por ejemplo, al código que llamó la función).
5. **Restart:**
   Reinicia toda la depuración desde el principio del programa.
   Muy útil cuando se quiere volver a repetir la ejecución después de modificar el código.

6. **Stop:**
   Detiene completamente la depuración. El programa deja de ejecutarse y se restablece el entorno.

F. Consola de Depuración (Debug Console)

1. **Ubicación:**
   En la parte inferior de VS Code se encuentra un panel con pestañas: **Problems, Output, Debug Console y Terminal**.
   Selecciona **Debug Console** para acceder a la consola de depuración.
2. **Funciones principales:**
   - Muestra la salida de los comandos `console.log()` (en JavaScript) o `print()` (en otros lenguajes).
   - Permite escribir expresiones o variables para ver sus valores durante la ejecución.
   - Es una herramienta interactiva muy útil para verificar resultados sin modificar el código.
3. **Ejemplo de uso:**
   - Coloca un **segundo breakpoint** dentro de la función `clickeado`.
   - Ejecuta el depurador y haz clic en el botón **“cliquéame”** desde el navegador.
   - El código se detendrá justo en el breakpoint dentro de la función, permitiendo observar los valores de las variables relacionadas.
4. **Concepto clave:**
   Las líneas de código **no se ejecutan** hasta que el depurador las alcanza.
   Si usas **Step Over**, verás cómo la ejecución avanza línea por línea, mostrando cómo cambian los valores de las variables.
   Esta visualización permite entender **exactamente qué hace el programa en cada paso**.

Conclusión

La depuración en VS Code es una herramienta extremadamente poderosa y visual.
Permite inspeccionar variables, seguir la ejecución paso a paso y comprender cómo fluye el código dentro de una aplicación.
Dominar el depurador no solo ayuda a resolver errores más rápido, sino que también mejora tu capacidad para **leer y comprender código complejo**.

Gracias a su integración con múltiples lenguajes y navegadores, el depurador de VS Code se adapta tanto a principiantes como a profesionales que buscan optimizar su flujo de trabajo de desarrollo.

X. Extensiones y Temas

Visual Studio Code (VS Code) es una herramienta extremadamente poderosa no solo por sus funciones integradas, sino también por su capacidad de **personalización a través de extensiones y temas**.
Gracias a ellas, cada desarrollador puede adaptar su entorno de trabajo a su estilo, necesidades y estética personal. Esta lección explica cómo funcionan las extensiones, cómo cambiar el aspecto visual de VS Code y cómo instalar temas e iconos que mejoren la experiencia de programación.

A. Funcionalidad de Extensiones

1. **Extender las funcionalidades de VS Code**
   Las extensiones son complementos o "miniaplicaciones" que amplían las capacidades nativas de VS Code.
   Con ellas se pueden añadir nuevas herramientas, soportes de lenguajes, depuradores, linters, emuladores de terminal, analizadores de código, integración con bases de datos y mucho más.

   Por ejemplo:

   - Instalar soporte para **Python**, **C#**, **Java** o **HTML/CSS/JS**.
   - Añadir un **formateador automático** (como *Prettier*).
   - Integrar **GitHub Copilot** o **ChatGPT** para asistencia con código.
   - Instalar extensiones de productividad (como *Live Server* para ejecutar proyectos web en tiempo real).

   En definitiva, las extensiones transforman VS Code en un entorno de desarrollo completo y adaptable a cualquier lenguaje o tecnología.

2. **Personalizar la apariencia de VS Code (Look & Feel)**
   Además de mejorar sus funciones, las extensiones también permiten modificar el aspecto visual del editor: colores, tipografías, íconos, sombras, fondos, animaciones, etc.
   Personalizar la interfaz no solo es cuestión de estética: un entorno visual cómodo y atractivo mejora la concentración, reduce la fatiga ocular y hace más placentera la experiencia de programar.
   En esta parte exploraremos los **temas de color** y los **paquetes de íconos**, dos de los tipos de extensiones más populares.

B. Temas

Los **temas** cambian la paleta de colores general de VS Code: el fondo, el color de la fuente, los tonos de las etiquetas, palabras clave, comentarios y el resaltado del cursor.

1. **Acceso a la tienda de temas**
   Para instalar un tema, sigue estos pasos:

   - Haz clic en el **icono de extensiones** (símbolo de cuadrado dividido en cuatro) en la barra de actividades, al lado izquierdo de la pantalla.

- En la caja de búsqueda escribe el nombre del tema que deseas instalar o simplemente “theme” para explorar las opciones disponibles.
- Verás una lista con cientos de temas disponibles. Cada uno tiene una vista previa, calificación y número de descargas.

2. **Sugerencias de temas populares**
   A continuación, una lista con algunos de los temas más recomendados y utilizados por la comunidad:

   a. **Andrómeda**
   Inspirado en la galaxia del mismo nombre, es un tema moderno, elegante y lleno de contrastes.
   Incluye variantes como *Andrómeda Border, Andrómeda Italic* o *Andrómeda Default*.
   Su combinación de tonos violetas y azules profundos crea una atmósfera de trabajo inmersiva y profesional.

   b. **Code Blue**
   Un tema limpio, con un fondo azul oscuro y colores suaves. Perfecto para quienes prefieren una apariencia sobria sin perder estilo.
   Su legibilidad es excelente, incluso en sesiones largas de programación nocturna.

   c. **Community Material Theme / Material Optima**
   Basado en las guías de diseño de Google Material, es uno de los temas más famosos y versátiles.
   Ofrece distintas variantes (Ocean, Palenight, Darker, Lighter, etc.), todas con una paleta cuidadosamente equilibrada para evitar fatiga visual.

   d. **Pain Night**
   Ideal para quienes buscan un ambiente oscuro y dramático. Su fondo negro profundo con tonos púrpura y gris oscuro crea un entorno elegante, artístico y relajante.

   e. **Groove Box**
   Diseñado para los nostálgicos de los terminales antiguos. Usa colores terrosos, ámbar y verdes desaturados, evocando la estética de las computadoras retro.
   Es perfecto si te gusta sentir que trabajas dentro de una terminal clásica.

   Una vez que instales un tema, se activará automáticamente, aunque puedes cambiarlo en cualquier momento.

C. Iconos

Los **paquetes de iconos** son extensiones que modifican los símbolos que aparecen junto a los archivos y carpetas del explorador de VS Code.
Cada tipo de archivo (HTML, CSS, JS, JSON, Python, imágenes, carpetas, etc.) tendrá un ícono distintivo, facilitando su identificación visual.

1. **Sugerencia popular: Material Icons**
   Uno de los paquetes de iconos más descargados es **Material Icon Theme**, creado siguiendo la estética de Google Material Design.
   Al instalarlo, los íconos por defecto de VS Code son reemplazados por versiones coloridas,

planas y muy reconocibles.
Por ejemplo:

- Los archivos **HTML** muestran un ícono naranja con la etiqueta “<>”.
- Los archivos **JavaScript** se representan con el logo amarillo “JS”.
- Las carpetas adoptan distintos colores según su tipo (src, assets, components, etc.).

Esta personalización no solo embellece el entorno, sino que **acelera la navegación visual** por los proyectos, ya que cada tipo de archivo puede identificarse instantáneamente.

D. Cambio de Tema Rápido

Además de instalar temas desde la vista de extensiones, VS Code ofrece una manera más rápida de cambiar el tema activo mediante la **Paleta de Comandos**.

1. **Abrir la Paleta de Comandos**
   - Presiona las teclas **Shift + Control + P**.
   - Aparecerá una barra superior donde puedes escribir comandos para realizar acciones dentro del editor.
2. **Buscar el comando de temas**
   Escribe la palabra **theme**.
   VS Code mostrará una lista de opciones relacionadas con la apariencia.
3. **Seleccionar “Preferences: Color Theme”**
   Escoge la opción **“Preferences: Color Theme”** (también se puede acceder directamente con el atajo **Control + K** y luego **Control + T**).
   Al hacerlo, se desplegará una lista con todos los temas instalados.
4. **Seleccionar el tema deseado**
   Puedes desplazarte por la lista usando las flechas del teclado para obtener una **vista previa inmediata** del tema.
   Una vez encuentres el que más te guste, presiona **Enter** para aplicarlo definitivamente.

   Si no te convence, repite el proceso hasta encontrar la combinación perfecta que te inspire a trabajar cómodamente.

Conclusión

Las extensiones y temas convierten a VS Code en una herramienta **viva y adaptable**.
No se trata solo de programar, sino de crear un entorno que refleje tu identidad como desarrolladora o creadora digital.
Gracias a las extensiones, puedes añadir soporte para cualquier lenguaje, automatizar tareas, mejorar la productividad y transformar la estética visual del editor hasta hacerlo único.

Dominar este apartado te permitirá tener un entorno de desarrollo **eficiente, motivador y visualmente inspirador**, algo esencial para disfrutar cada sesión de trabajo.

XI. Atajos de Teclado y Edición Rápida

Visual Studio Code (VS Code) es muy poderoso gracias a sus **atajos de teclado**, que permiten navegar, editar y manipular el código de manera rápida y eficiente.
Dominar estos atajos no solo acelera tu flujo de trabajo, sino que reduce la fatiga al evitar depender constantemente del ratón. Esta lección explica los atajos más importantes para **navegación, manipulación de líneas, indentación y comentarios**.

A. Navegación

La navegación rápida permite moverse por el código sin perder el foco ni interrumpir la concentración.

1. **Ir al final de línea:**
    - Windows: **Fin**
    - Mac: **Comando + Derecha**
      Esto mueve el cursor directamente al final de la línea actual, útil para agregar código al final de instrucciones o comentarios.
2. **Ir al inicio de línea:**
    - Windows: **Inicio**
    - Mac: **Comando + Izquierda**
      Permite colocar el cursor al inicio de la línea para editar o insertar código al principio.
3. **Moverse entre palabras:**
    - Teclas: **Alt + Izquierda/Derecha**
      Saltar de palabra en palabra en lugar de carácter por carácter agiliza la navegación en líneas largas de código.
4. **Seleccionar texto por palabras:**
    - Teclas: **Alt + Shift + Izquierda/Derecha**
      Esto permite **seleccionar rápidamente palabras completas** sin tener que mantener presionado Shift y mover el cursor manualmente.
      Ideal para copiar, cortar o reemplazar palabras enteras sin perder tiempo.

B. Manipulación de Líneas

VS Code permite mover líneas completas sin necesidad de cortar y pegar, lo cual es extremadamente útil para reorganizar código.

1. **Subir o bajar una línea:**
    - Teclas: **Alt + Arriba/Abajo**
      Esto intercambia la línea actual con la línea de arriba o abajo manteniendo el cursor en la misma posición relativa.

- Ejemplo: Mover la declaración de una variable a otra sección del código sin cortar y pegar manualmente.

Esta funcionalidad también funciona seleccionando varias líneas: todas se moverán juntas como un bloque.

C. Indentación

La indentación correcta es fundamental en cualquier lenguaje de programación, no solo para la **legibilidad**, sino también para la **ejecución correcta**, especialmente en Python.

1. **Indentar a la derecha:**
    - Tecla: **Tab**
      Desplaza la línea o bloque de líneas seleccionadas un nivel hacia la derecha.
2. **Indentar a la izquierda:**
    - Tecla: **Shift + Tab**
      Mueve la línea o bloque de líneas hacia la izquierda, disminuyendo el nivel de indentación.

Usar estos atajos es mucho más rápido que hacerlo manualmente con espacios y asegura una **indentación consistente** en todo el archivo.

D. Comentarios

Comentar código es una práctica esencial para documentar, probar o deshabilitar temporalmente partes de un programa. VS Code facilita esta tarea con atajos rápidos.

1. **Comentar una línea:**
    - Teclas: **Control** + **/** (Windows)
    - Mac: **Comando** + **/**
      Esto agrega automáticamente el símbolo de comentario correspondiente al lenguaje actual (por ejemplo, `//` en JavaScript o # en Python).
2. **Comentar múltiples líneas:**
    - Selecciona varias líneas y presiona **Control** + **/**
      Cada línea seleccionada se comentará automáticamente, respetando la sintaxis del lenguaje.
3. **Descomentar:**
    - Usa el mismo atajo **Control** + **/** sobre líneas ya comentadas.
      VS Code detecta que están comentadas y elimina los símbolos de comentario, restaurando el código a su estado original.
4. **Buscar atajos en el menú:**

- Menú superior: **Edit → Toggle Line Comment**
  Esta opción permite realizar la misma acción de comentar/descomentar usando el ratón si no recuerdas el atajo, aunque el atajo de teclado es mucho más rápido.

Conclusión

Los **atajos de teclado y la edición rápida** son herramientas esenciales para cualquier desarrollador que quiera ser productivo en VS Code.

- La **navegación rápida** permite moverse por el código sin perder foco.
- La **manipulación de líneas** agiliza la reestructuración de código.
- La **indentación** asegura un estilo consistente y evita errores de ejecución.
- Los **comentarios** facilitan la documentación y pruebas rápidas del código.

Dominar estos atajos convierte a VS Code en un entorno ágil y profesional, donde la velocidad y eficiencia aumentan considerablemente, permitiendo concentrarse en lo realmente importante: **programar y crear soluciones**.

XII. IntelliSense y Snippets

Visual Studio Code (VS Code) no solo es un editor de código, sino también un **asistente inteligente** que facilita la escritura y corrección del código mediante **IntelliSense** y **snippets**.
Estas herramientas permiten **escribir código más rápido**, **reducir errores** y **aumentar la productividad**, especialmente en proyectos grandes o al trabajar con frameworks complejos.

A. IntelliSense

1. **Función principal**
   IntelliSense es la característica que sugiere automáticamente **métodos, propiedades y parámetros** de objetos o clases mientras escribes.

   - Ejemplo: si escribes

     `miString.`

     VS Code mostrará todas las funciones disponibles para ese objeto, como `.length`, `.toUpperCase()`, `.trim()`, etc.

   - Otro ejemplo:

     `console.`

     IntelliSense mostrará métodos como `log`, `error`, `warn`, `table`, entre otros.

2. **Navegación de sugerencias**

   - Usa las **flechas arriba/abajo** para moverte entre las opciones de IntelliSense.
   - Presiona **Enter** para seleccionar la opción deseada.
   - Puedes también presionar **Tab** para autocompletar la sugerencia seleccionada.

3. **Beneficios**

   - Reduce errores de escritura y sintaxis.
   - Permite descubrir métodos y propiedades de objetos sin necesidad de consultar documentación externa.
   - Acelera la programación al evitar escribir largas cadenas de código manualmente.

B. Creación de Snippets (Fragmentos de código)

Los **snippets** son fragmentos de código predefinidos que se pueden insertar automáticamente escribiendo un **prefix** (palabra clave corta).
Sirven para **automatizar la escritura de código repetitivo** y mantener consistencia en la sintaxis.

1. **Propósito**

   - Permitir escribir un par de letras o palabra corta y generar automáticamente un bloque de código completo.
   - Ejemplo: en lugar de escribir `console.log()` cada vez, puedes usar un snippet que al escribir `clg` genere todo el comando.

2. **Acceso a la creación de snippets**
    - Haz clic en el **engranaje** en la esquina inferior izquierda → **Configure User Snippets**.
    - Selecciona el archivo de lenguaje correspondiente, por ejemplo `javascript.json` para snippets de JavaScript.
3. **Configuración de un snippet**

    Cada snippet tiene tres componentes principales:

    a. Prefix:
    - Es la palabra clave que gatilla el snippet.
    - Ejemplo: `clg`.

    b. Body:
    - Es el contenido que se insertará en el editor.
    - Ejemplo de snippet para `console.log()`:

```
"Console Log": {
  "prefix": "clg",
  "body": [
    "console.log($1);"
  ],
  "description": "Imprime en consola"
}
```

    - **Uso de $1, $2, …**:
        - `$1` indica la primera posición donde estará el cursor después de insertar el snippet.
        - Presionando **Tab** se mueve a la siguiente posición `$2`, `$3`, etc., facilitando la edición rápida dentro del fragmento.

    c. Description:
    - Texto que explica el propósito del snippet.
    - Aparece como referencia cuando IntelliSense sugiere el snippet.
    - Ejemplo: `"description": "Imprime en consola"`
4. **Uso del snippet**
    - Escribe el **prefix** en el editor (`clg`).
    - Presiona **Enter** o selecciona la sugerencia de IntelliSense.
    - VS Code reemplazará el prefix por el **código completo** definido en `body`.

C. Snippets de Terceros

1. **Instalación desde el Marketplace**
    - VS Code permite descargar extensiones de snippets creados por la comunidad o empresas.
    - Ejemplo muy popular: **JavaScript ES6 Snippets**, con más de 9.5 millones de descargas.
    - Estas extensiones incluyen **snippets para funciones, imports, clases y patrones comunes**.
2. **Snippets para frameworks y librerías**
    - Puedes encontrar paquetes específicos para:
        - **React:** incluye snippets para componentes funcionales, hooks, props y estados.
        - **Angular:** genera componentes, servicios, módulos y decoradores rápidamente.
        - **Vue:** snippets para `template`, `script`, `style` y directivas de Vue.
3. **Ventajas de usar snippets de terceros**
    - Acelera la escritura de código repetitivo.
    - Garantiza consistencia en la estructura del proyecto.
    - Permite aprender patrones de código recomendados por la comunidad.

Conclusión

El uso de **IntelliSense y Snippets** en VS Code **potencia la productividad y reduce errores** al programar.

- IntelliSense permite **navegar y autocompletar métodos** y propiedades de manera inteligente.
- Los snippets personalizados o de terceros permiten **generar rápidamente bloques de código repetitivo**.

Dominar estas herramientas convierte a VS Code en un **entorno de desarrollo ágil, eficiente y profesional**, donde escribir código se vuelve mucho más rápido y organizado.

XIII. Navegación Rápida y Paleta de Comandos

Visual Studio Code (VS Code) no solo es un editor de código moderno y potente, sino también un entorno diseñado para **la máxima eficiencia en la navegación y ejecución de tareas**.
Entre sus herramientas más importantes se encuentran la **búsqueda rápida de archivos** y la **Paleta de Comandos**, que permiten interactuar con el editor sin depender del ratón, aumentando notablemente la productividad.

A. Búsqueda de Archivos ($Control+P$)

1. **Función principal**
    - Esta funcionalidad permite buscar cualquier archivo dentro del proyecto escribiendo **parte del nombre o una cadena de caracteres**.
    - Es extremadamente útil en proyectos grandes con cientos o miles de archivos, donde navegar manualmente por carpetas sería ineficiente.
2. **Cómo usarla**
    - Presiona **Control + P** (Windows/Linux) o **Comando + P** (Mac).
    - Aparecerá una barra de búsqueda en la parte superior del editor.
    - Escribe parte del nombre del archivo que deseas abrir, por ejemplo: `index`, `readme` o incluso fragmentos más específicos.
    - VS Code mostrará en tiempo real una lista de archivos que coinciden con los caracteres ingresados.
    - Presiona **Enter** para abrir el archivo seleccionado.
3. **Ventajas**
    - **Ahorro de tiempo:** evita navegar manualmente por carpetas profundas.
    - **Precisión:** permite encontrar archivos incluso cuando no recuerdas la ruta completa.
    - **Flexibilidad:** se pueden buscar no solo nombres exactos, sino también fragmentos de nombres, gracias a la coincidencia parcial.

B. Paleta de Comandos ($Shift+Control+P$)

1. **Función principal**
    - La Paleta de Comandos es **la barra maestra de VS Code**, que permite ejecutar **cualquier acción disponible en el editor**, sin importar si normalmente se realiza desde menús, botones o atajos de teclado.
    - Con ella, puedes: guardar archivos, comentar líneas, abrir carpetas, cambiar temas, instalar extensiones, ejecutar depuraciones y mucho más.
2. **Cómo abrir la Paleta de Comandos**
    - Presiona **Shift + Control + P** (Windows/Linux) o **Shift + Comando + P** (Mac).

- Aparecerá una barra en la parte superior del editor con un cuadro de búsqueda donde puedes escribir la acción que deseas realizar.

3. **Ejemplos de uso**

   a. Comentar líneas

   - Escribe `comment` en la barra.
   - Selecciona **Toggle Line Comment**.
   - Esto comentará o descomentará las líneas seleccionadas sin necesidad de memorizar atajos de teclado.

   b. Guardar archivo

   - Escribe `save`.
   - Selecciona **File: Save**.
   - Esto guarda el archivo activo, replicando la acción de **Control + S**.

4. **Características adicionales**
   - Permite **buscar cualquier comando** aunque no recuerdes el atajo correspondiente.
   - La lista de sugerencias se actualiza dinámicamente mientras escribes, mostrando acciones relacionadas.
   - Puedes ejecutar comandos de **extensiones instaladas**, configuraciones avanzadas o tareas automáticas definidas por ti.
5. **Ventajas de la Paleta de Comandos**
   - **Centralización:** un solo lugar desde donde realizar cualquier acción del editor.
   - **Aprendizaje:** facilita descubrir funciones avanzadas de VS Code sin necesidad de navegar por menús.
   - **Productividad:** reduce la dependencia del ratón y agiliza el flujo de trabajo.

Conclusión

La **búsqueda rápida de archivos** y la **Paleta de Comandos** son herramientas esenciales para trabajar de manera ágil en VS Code.

- Con **Control + P**, puedes abrir cualquier archivo en segundos, incluso en proyectos extensos.
- Con **Shift + Control + P**, tienes acceso a **todas las funciones del editor**, lo que lo convierte en un entorno altamente eficiente y adaptable a cualquier flujo de trabajo.

Dominar estas herramientas permite **minimizar interrupciones, maximizar la productividad y mantener el foco en la programación**, aprovechando al máximo las capacidades de VS Code.

XIV. Personalización del Layout de VS Code

Visual Studio Code (VS Code) no solo es potente por sus funcionalidades, sino también por su **flexibilidad visual**, permitiendo que cada desarrollador adapte el entorno a su estilo de trabajo.
El layout (diseño de la interfaz) puede configurarse para **optimizar el espacio de código, reducir distracciones y mejorar la productividad**. Esta lección detalla cómo personalizar el layout paso a paso.

A. Mover la Barra Lateral Primaria (Sidebar)

La **barra lateral** o sidebar contiene elementos esenciales como el Explorador, Control de Versiones, Debugger y Extensiones.
Moverla puede mejorar la experiencia visual, evitando que el código se desplace demasiado al abrir paneles.

1. **Acceso a la configuración de layout**
    - Ir al menú superior: **View → Appearance → Customize Layouts**
2. **Seleccionar posición de la barra lateral primaria**
    - En el menú de personalización, elegir: **Primary Side Bar Position**
3. **Mover la barra lateral a la derecha**
    - Seleccionar **Right**
    - Esta configuración es útil para quienes prefieren ver el código principalmente a la izquierda y tener los paneles de apoyo en el lado derecho, evitando que el código se reduzca al abrir el sidebar.

B. Ocultar la Barra de Actividades (Activity Bar)

La **Activity Bar** es la barra vertical que contiene los iconos de **Explorer, Buscar, Git, Debug y Extensiones**.
Si deseas un entorno más minimalista o concentrarte en el código, puedes ocultarla:

1. **Acceso**
    - Mismo menú: **View → Appearance → Customize Layouts**
2. **Ocultar Activity Bar**
    - Selecciona **Activity Bar**
    - La barra desaparecerá del lateral, dejando más espacio horizontal para el código.

C. Alternar visibilidad de la barra lateral

Si prefieres ocultar temporalmente la barra lateral para maximizar el área de edición:

1. **Atajo de teclado**
    - Presiona **Control + B** (Windows/Linux) o **Comando + B** (Mac)

- Esta acción **oculta o muestra la barra lateral** según su estado actual.

2. **Ventajas**
    - Permite concentrarte en el código sin eliminar permanentemente los paneles.
    - Ideal para pantallas pequeñas o laptops, donde cada pixel de espacio cuenta.

D. Ocultar pestañas (Tabs)

Cuando trabajas con muchos archivos, las pestañas pueden saturar la parte superior del editor. VS Code permite ocultarlas para una vista más limpia:

1. **Acceso a Configuración**
    - Menú: **File → Preferences → Settings**
    - Atajo: **Control + ,**
2. **Buscar la opción de pestañas**
    - Escribe **tabs** en la barra de búsqueda de configuración
3. **Desactivar la visualización de pestañas**
    - Busca **Workbench: Editor: Show Tabs**
    - Desactiva la opción
    - Ahora los archivos abiertos no mostrarán pestañas, pero seguirán accesibles a través de la navegación de archivos abiertos.

E. Navegación entre archivos abiertos sin pestañas

Cuando se ocultan las pestañas, VS Code ofrece métodos rápidos para moverse entre archivos abiertos:

1. **Atajo de teclado**
    - Presiona **Control + K** y luego **P** (Windows/Linux)
    - Mac: **Comando + K, luego P**
2. **Funcionalidad**
    - Se despliega una lista de archivos abiertos actualmente
    - Puedes seleccionar con las flechas del teclado y presionar **Enter** para cambiar de archivo
    - Esto reemplaza la función visual de las pestañas y mantiene un flujo de trabajo ágil

---

Conclusión

La **personalización del layout** en VS Code permite crear un entorno de trabajo cómodo, eficiente y libre de distracciones:

- Mover la **barra lateral** a la derecha mejora la visualización del código.
- Ocultar la **Activity Bar** o las **pestañas** reduce el desorden visual.
- Atajos como **Control + B** y la navegación de archivos abiertos permiten mantener la eficiencia incluso sin pestañas visibles.

Dominar estas opciones ayuda a **adaptar VS Code a tu estilo personal**, optimizando la productividad y la comodidad durante largas sesiones de programación.

XV. Atajos de Teclado Customizados

Visual Studio Code (VS Code) permite **personalizar atajos de teclado**, lo que facilita adaptar el editor a tu flujo de trabajo y ganar eficiencia al realizar tareas repetitivas.
En esta lección se explica cómo **enfocar el explorador**, navegar rápidamente entre archivos y **crear atajos customizados**, incluyendo un ejemplo práctico para **crear un nuevo archivo con la tecla 'a'**.

A. Enfocar el Explorador

El **Explorer** (Explorador de Archivos) es fundamental para navegar entre los archivos de un proyecto. VS Code permite centrar el foco en él mediante un atajo de teclado:

1. **Atajo de teclado para enfocar el Explorer**
    - Windows/Linux: **Shift + Control + E**
    - Mac: **Shift + Comando + E**
    - Al presionar este atajo, el Explorador resalta el archivo actualmente abierto o seleccionado.
2. **Moverse entre archivos**
    - Una vez enfocado el Explorador, usa las **flechas arriba/abajo** para seleccionar archivos o carpetas.
3. **Previsualización de archivos**
    - Presiona **Espacio** mientras un archivo está seleccionado en el Explorer.
    - Esto abre una **vista previa temporal**, útil para revisar contenido sin abrir completamente el archivo en una pestaña.
4. **Beneficios**
    - Navegación rápida entre archivos sin tocar el ratón.
    - Posibilidad de abrir y explorar archivos grandes de manera eficiente.

B. Creación de un Atajo Customizado: Crear Nuevo Archivo con la tecla 'a'

Una de las grandes ventajas de VS Code es que puedes **asignar cualquier combinación de teclas** a funciones específicas, incluso con condiciones de contexto.
A continuación, un ejemplo práctico para crear un **nuevo archivo en el Explorador usando solo la tecla 'a'**.

1. **Acceso a los atajos de teclado**
    - Menú: **File → Preferences → Keyboard Shortcuts**
    - Atajo: **Control + K, luego Control + S**
    - Esto abre el editor de **Keyboard Shortcuts** donde se pueden buscar, modificar y crear atajos.

2. **Buscar la acción a customizar**
    - En la barra de búsqueda, escribe: **New file**
    - Se mostrará la acción correspondiente a crear un archivo nuevo en el Explorador.
3. **Seleccionar la acción correcta**
    - Asegúrate de seleccionar la acción que **opera dentro del Explorer**, no otra función de "nuevo archivo" que pueda estar en menús contextuales distintos.
4. **Asignar la tecla**
    - Haz clic en el icono de lápiz junto a la acción y presiona **a** como nuevo atajo.
5. **Establecer condiciones (When Expression)**
    - Esto asegura que la tecla 'a' solo funcione **cuando el foco esté en el Explorer** y no afecte otras funciones del editor:
        - `explorerViewletVisible` → El explorador está visible.
        - `filesExplorerFocus` → El foco está dentro del explorador.
        - `!explorerResourceReadOnly` → El archivo no es de solo lectura.
        - `!inputFocus` → No se está escribiendo en un input de búsqueda u otra caja de texto.
6. **Prueba del atajo**
    - Presiona **Shift + Control + E** para enfocar el Explorer.
    - Presiona **a**
    - Aparecerá el campo para nombrar un **nuevo archivo**, por ejemplo: `HolaMundo.js`.
    - Presiona **Enter** y el archivo se creará automáticamente en la carpeta seleccionada.

Conclusión

Los **atajos de teclado customizados** en VS Code permiten:

- Optimizar tareas repetitivas.
- Adaptar el entorno a tu flujo de trabajo personal.
- Incrementar la velocidad y eficiencia en la navegación y edición de proyectos grandes.

El ejemplo mostrado demuestra cómo una acción cotidiana, como **crear un archivo**, puede reducirse a **una sola tecla**, manteniendo el foco en la productividad y reduciendo la dependencia del ratón.

XVI. Configuración de la Terminal Integrada

La **Terminal Integrada** de Visual Studio Code (VS Code) es una herramienta fundamental para cualquier desarrollador, ya que permite ejecutar comandos, scripts y control de versiones sin salir del editor.
Configurar correctamente la terminal integrada mejora la **fluidez del trabajo**, optimiza el **uso de comandos de Git, Node.js o Python**, y proporciona una experiencia visual más cómoda.

A. Cambio de Terminal por Defecto

Por defecto, VS Code utiliza **PowerShell** en Windows, que muchos usuarios consideran visualmente limitado o “feo”. Se recomienda cambiarla por una terminal más amigable, como **Git Bash** o **WSL (Windows Subsystem for Linux)** si se usa Linux dentro de Windows.

1. **Abrir la terminal integrada**
    - Menú: **View → Terminal**
    - Esto abrirá la terminal inferior con el perfil por defecto (PowerShell).
2. **Seleccionar la terminal por defecto**
    - Haz clic en la **flecha hacia abajo** junto al icono **+** de la terminal.
    - Selecciona: **Select Default Profile**
3. **Elegir Git Bash o WSL**
    - En la lista de perfiles disponibles, selecciona:
        - **Git Bash:** recomendada para usuarios de Git y comandos Unix.
        - **WSL:** si tienes instalado el Subsistema de Linux en Windows.
    - VS Code configurará esta terminal como predeterminada para futuras sesiones.
4. **Ventajas de Git Bash/WSL sobre PowerShell**
    - Mejor compatibilidad con scripts de Bash.
    - Acceso a comandos Unix/Linux comunes (`ls`, `cat`, `grep`, etc.).
    - Mejor visualización de colores y prompts.
    - Integración fluida con Git para control de versiones.

B. Abrir la Terminal Integrada

Una vez configurada la terminal por defecto, es importante saber abrirla rápidamente sin usar el ratón:

1. **Atajo de teclado**
    - Windows/Linux: **Control** + ` (la tecla de tilde, justo debajo de Esc)
    - Mac: **Control** + **`** o **Comando** + según configuración

2. **Funcionamiento**
    - Abre la terminal integrada con el perfil por defecto seleccionado (Git Bash en este caso).
    - Puedes tener varias terminales abiertas simultáneamente:
        - Botón **+** para abrir nuevas pestañas de terminal.
        - Menú desplegable para cambiar entre perfiles si necesitas otra terminal temporalmente.
3. **Ventajas de usar la terminal integrada**
    - Ejecutar scripts sin salir del editor.
    - Integración directa con Git y depuradores.
    - Permite ver errores, advertencias y resultados de comandos de manera inmediata.
    - Mantiene el flujo de trabajo centralizado dentro de VS Code, evitando cambiar entre ventanas externas.

Conclusión

Configurar correctamente la **terminal integrada** en VS Code es un paso clave para un flujo de trabajo profesional:

- Cambiar la terminal por defecto a **Git Bash o WSL** mejora la compatibilidad y la estética.
- Abrir la terminal con **Control +** ` permite un acceso rápido y eficiente.
- Tener la terminal integrada facilita la ejecución de scripts, manejo de Git y depuración sin interrumpir la edición de código.

Dominar esta configuración asegura **una experiencia de desarrollo más fluida, productiva y centrada** dentro de VS Code.

XVII. Conclusión

Hemos llegado al **final de este curso completo de Visual Studio Code (VS Code)**. Esta lección está diseñada para **resumir lo aprendido**, consolidar los conocimientos y ofrecer pasos finales para continuar creciendo como desarrollador.

A. Finalización del curso y logros

1. **Resumen de lo aprendido**
   A lo largo de las lecciones, hemos cubierto desde la instalación hasta el uso avanzado de VS Code:
   - **Introducción y preparación:** comprendiste la importancia de VS Code, los requisitos y cómo prepararte para un aprendizaje eficiente.
   - **Descarga e instalación:** aprendiste a instalar VS Code en Windows y explorar su primera vista.
   - **Configuración inicial:** cambiaste el idioma y ajustaste la interfaz a tus preferencias.
   - **Componentes principales:** dominaste la barra de actividades, barra de estado y menú superior.
   - **Explorador de archivos:** creaste, editaste y organizaste archivos y carpetas de manera profesional.
   - **Barra de estado y búsqueda:** aprendiste a interpretar indicadores, navegar por líneas y realizar búsquedas y reemplazos de texto.
   - **Control de versiones con Git:** instalaste Git, inicializaste repositorios, manejaste ramas y conectaste con GitHub.
   - **Depuración:** configuraste breakpoints, controlaste la ejecución y analizaste variables en la consola de depuración.
   - **Extensiones y temas:** personalizaste VS Code con temas, iconos y snippets.
   - **Atajos de teclado y edición rápida:** optimizaste tu flujo de trabajo con combinaciones de teclas y navegación eficiente.
   - **IntelliSense y snippets:** aprendiste a autocompletar código y crear fragmentos personalizados.
   - **Paleta de comandos y navegación rápida:** accediste a todas las funciones del editor sin usar el ratón.
   - **Personalización del layout y terminal integrada:** adaptaste el espacio de trabajo a tus necesidades y configuraste la terminal por defecto.
   - **Atajos customizados:** creaste tus propias combinaciones de teclas para tareas repetitivas y aumentaste la productividad.
2. **Logros alcanzados**
   - Capacidad de **configurar VS Code** completamente a tu gusto.

- Habilidad para **navegar, editar, depurar y versionar código** de forma profesional.
- Dominio de **atajos, snippets y terminal integrada**, permitiendo un flujo de trabajo ágil y eficiente.
- Comprensión de la **importancia de la personalización y optimización** del entorno de desarrollo.

Con estos conocimientos, ya estás listo para trabajar de manera profesional en proyectos personales o colaborativos, y para **aprovechar al máximo VS Code como tu herramienta principal de programación**.

B. Agradecimiento y llamados a la acción

1. **Agradecimiento**
    - Gracias por tu tiempo, constancia y dedicación durante este curso.
    - Aprender a dominar un editor de código tan potente como VS Code **marca la diferencia** en tu desempeño como desarrollador.
2. **Llamados a la acción**
    - **Me gusta:** si este curso te resultó útil, demuestra tu apoyo presionando "Me gusta".
    - **Suscribirse:** suscríbete para recibir más cursos, tutoriales y contenido de programación avanzado.
    - **Comentarios:** comparte tus dudas, sugerencias o experiencias con VS Code; tu retroalimentación ayuda a mejorar futuros cursos.
    - **Explora otros cursos:** sigue aprendiendo sobre Git, JavaScript, Python, frameworks modernos o desarrollo web y móvil.

Cierre motivacional

Dominar VS Code es **más que aprender a usar un editor**: es **desarrollar hábitos de productividad, organización y eficiencia** que te acompañarán durante toda tu carrera como programador.

Recuerda: la **práctica constante** es clave. Aplica lo aprendido en proyectos reales, explora nuevas extensiones, ajusta tu layout y experimenta con atajos para encontrar tu flujo de trabajo ideal.

VS Code es tu **herramienta, tu espacio creativo y tu asistente**: cuanto mejor la conozcas, más rápido y seguro será tu desarrollo profesional.

Guía Rápida de VS Code

Sección	Funcionalidad / Comando	Atajo / Ruta	Notas / Tips
Explorador de Archivos	Abrir explorador	`Shift + Ctrl + E`	Resalta el archivo actual
	Crear archivo nuevo	Click botón + / Atajo customizable (`a`)	Funciona cuando el foco está en el Explorer
	Crear carpeta	Icono de carpeta +	Mantener organizado el código
	Colapsar todas carpetas	Icono colapsar	Útil en proyectos grandes
Pestañas y Layout	Ocultar pestañas	File → Preferences → Settings → Workbench: Editor: Show Tabs = Desactivar	Usar navegación por archivos abiertos (`Ctrl + K`, luego `P`)
	Alternar barra lateral	`Ctrl + B`	Oculta/muestra Explorer y Activity Bar
	Mover barra lateral	View → Appearance → Customize Layouts → Primary Side Bar Position → Right	Mejora visibilidad de código
Barra de Estado	Posición del cursor	Barra de estado inferior	`Ln` y `Col` muestran línea y columna
	Identación	Barra de estado → elegir espacios o tabs	2 o 4 espacios según convención
	Codificación y EOL	Barra de estado → UTF-8 / CRLF o LF	Evita conflictos en equipos multiplataforma
	Tipo de archivo	Barra de estado → seleccionar lenguaje	Markdown, JS, HTML, etc.
Búsqueda y Reemplazo	Abrir búsqueda	Icono lupa	Buscar texto en proyecto
	Opciones	Match Case, Whole Word	Respeta mayúsculas/minúsculas y palabras exactas
	Reemplazo	Caja Replace	Sustituye texto en todos los resultados
Control de Versiones (Git)	Inicializar repositorio	Control de versiones → Init	Agrega archivos al stage con +
	Commit	Caja de mensaje + botón Commit	Configurar `user.name` y `user.email` si es la primera vez
	Ramas	Branch → Create Branch / Publish / Merge / Delete	Sincronizar con GitHub
	Conexión GitHub	Branch → Sign in	Autorizar Git Credential Manager
Depuración (Debugger)	Abrir Debugger	Icono insecto	Run and Debug, seleccionar navegador
	Breakpoints	Click al lado de número de	Detiene la ejecución para

Sección	Funcionalidad / Comando	Atajo / Ruta	Notas / Tips
		línea	inspeccionar variables
	Controles	Step Over (F10), Step Into, Step Out, Restart, Stop	Navegación precisa del código
	Debug Console	Panel inferior	Muestra `console.log` o print durante depuración
IntelliSense y Snippets	Autocompletado	Escribir objeto + `.`	Seleccionar con flechas + Enter
	Crear Snippet	Gear → Configure User Snippets → Lenguaje	Prefix, Body, Description, posiciones `$1, $2...`
	Snippets de terceros	Marketplace	Ej: JavaScript ES6 Snippets
Paleta de Comandos	Abrir	`Shift + Ctrl + P`	Ejecuta cualquier comando sin ratón
	Ejemplos	`save`, `comment`	Comentar, guardar, cambiar configuración
Atajos de teclado generales	Navegación	Fin / Inicio / Alt + Izq/Dcha	Mover cursor al inicio/fin o por palabras
	Selección por palabra	Alt + Shift + Izq/Dcha	Selección precisa de bloques
	Mover líneas	Alt + Arriba/Abajo	Intercambia posición de líneas
	Indentación	Tab / Shift + Tab	Alinear código fácilmente
	Comentarios	Ctrl + /	Comentar/descomentar líneas o bloques
Terminal Integrada	Abrir terminal	`Ctrl + ~`	Usar perfil por defecto (Git Bash recomendado)
	Cambiar terminal por defecto	View → Terminal → Select Default Profile	Git Bash o WSL para experiencia Unix
	Múltiples terminales	`+` en panel terminal	Abrir varias sesiones simultáneamente
Extensiones y Temas	Instalar temas	Icono extensiones → buscar tema	Andrómeda, Pain Night, Code Blue, Groove Box
	Instalar iconos	Material Icons	Reemplaza iconos por defecto (HTML, JS, etc.)
	Cambiar tema rápido	`Shift + Ctrl + P` → theme → Preferences: Color Theme	Buscar y seleccionar tema deseado
Custom Atajos	Asignar atajo	File → Preferences → Keyboard Shortcuts	Atajo condicionado con "When Expression"

Tips Finales

1. **Mantén tus atajos organizados**: crea un backup de `keybindings.json` si configuras muchos custom shortcuts.
2. **Usa la terminal integrada**: evita cambiar entre ventanas, agiliza tu flujo de trabajo.

3. **Personaliza colores y temas**: reduce fatiga visual y mejora concentración.
4. **Aprovecha IntelliSense y snippets**: acelera la escritura de código y evita errores tipográficos.
5. **Practica constantemente**: la eficiencia real llega usando atajos, paleta de comandos y terminal todos los días.